




# Security in Software Engineering

**Eduardo Fernández-Medina**

[Eduardo.FdezMedina@uclm.es](mailto:Eduardo.FdezMedina@uclm.es)  
<http://alarcos.inf-cr.uclm.es/per/efmedina/>

Alarcos Research Group  
**University of Castilla-La Mancha, Spain**





## Aportaciones del Grupo Alarcos

Líneas de Seguridad abordadas

- Bases de Datos y Almacenes de Datos Seguros
- Seguridad en Documentos Multimedia y XML
- Seguridad en Aplicaciones basadas en Servicios Web
- Seguridad en el modelado de Procesos de Negocio
- Métricas de seguridad
- Modelos de Madurez de la Seguridad para PYMEs
- Ingeniería de requisitos de seguridad
- Seguridad en entornos de Grid Computing con dispositivos móviles

Eduardo Fernández-Medina – Security in Software Engineering

3

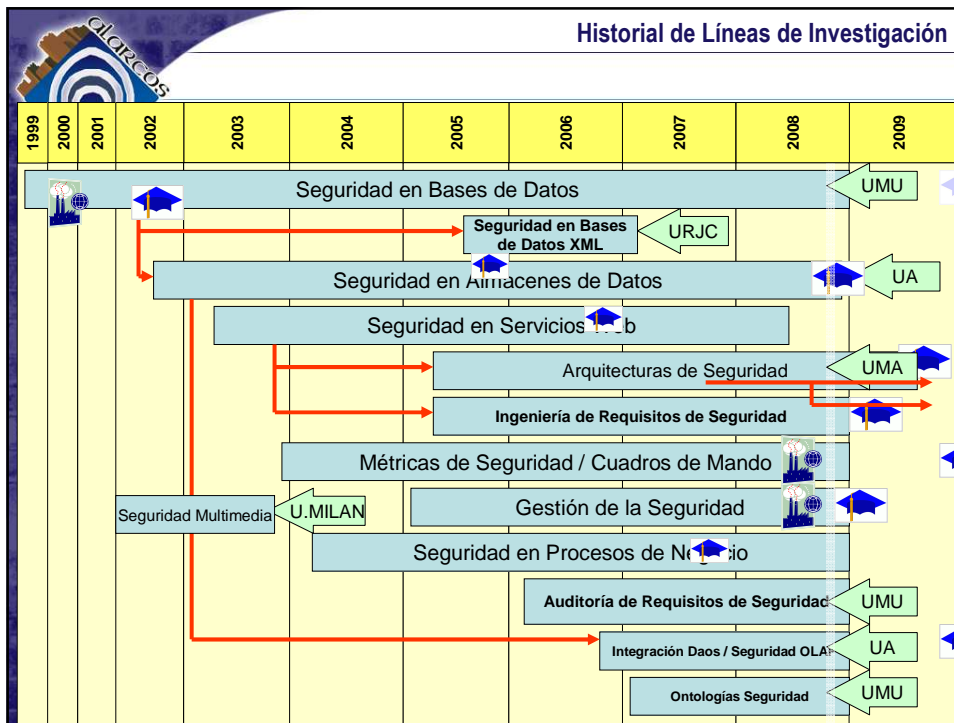


## Colaboraciones

Univ. Católica Maule Chile (Tesis)  
Univ. Bio Bio Chile (Tesis)  
Correos Telecom (Tesis)  
Univ. La Matanza Cuba (Tesis)  
Univ. Rey Juan Carlos (Investigación)  
Univ. Alicante (Investigación)  
Univ. Murcia (Investigación)  
Univ. Milán (Investigación)  
Univ. Carlos III (Organización Eventos)  
Univ. Complutense (Organización Eventos)  
Univ. Lleida (Organización Eventos)  
Univ. Polit. Cataluña (Organización Eventos)  
AENOR  
RETISTRUST

Eduardo Fernández-Medina – Security in Software Engineering

4



### Multidisciplinaridad

Seguridad Informática es un aspecto amplio.


Searching the Digital Bibliography & Library Project  
**FACETED DBLP**

**Browse authors:** [Most prolific \(overall\)](#) [Most prolific \(per year\)](#)  
**Browse conferences:** [Most prolific](#) [Longest running](#)  
**Browse journals:** [Most prolific](#) [Longest running](#)  
**Browse Keywords:** [Most popular](#)

Faceted DBLP - Browse Most Popular Keywords in DBLP++ - Windows Internet


<http://dblp.l3s.de/browse.php?browse=mostPopularKeywords>

6

Eduardo Fernández-Medina – Security in Software Engineering

**Security in Software Engineering**  
Introduction

**Most popular keywords**

This page lists the most popular author keywords assigned at least 100 times as available in our DBLP++ dataset.

Rank	Keyword name	Number of publications in DBLP
1	security	1591
2	data_mining	1540
3	clustering	1476
4	genetic_algorithms	1450
5	simulation	1447
6	scheduling	1432
7	Java	1428
8	QoS	1384
9	Internet	1344
10	XML	1236
11	performance_evaluation	1202
12	fault_tolerance	1145
13	ontology	1109
14	neural_networks	1072
15	optimization	1055
16	sensor_networks	1054
17	web_services	1035
18	information_retrieval	1029
19	World Wide Web (WWW)	1009
20	visualization	993
21	machine_learning	968
22	performance	959
23	wireless_sensor_networks	956
24	distributed_systems	936
25	modeling	932
26	software_engineering	916
27	classification	904
28	real-time systems	902

7


Eduardo Fernández-Medina – Security in Software Engineering

**Multidisciplinaridad**

- En la comunidad científica existen Centenares de Eventos relativos a seguridad (alrededor de 620 registrados):
  - <http://www.ieee-security.org/Calendar/cipher-hypercalendar.html>
- Los tópicos de investigación son “numerosos”
  - Authorization and Authentication
  - Availability and Reliability
  - Cost/Benefit Analysis
  - Cryptography
  - Dependability Aspects for Special Applications (e.g. ERP-Systems, Logistics)
  - Dependability Aspects of Electronic Government (e-Government)
  - Dependability Administration
  - Dependability in Open Source Software
  - Designing Security Requirements

8

Eduardo Fernández-Medina – Security in Software Engineering



**Multidisciplinaridad**

- Digital Forensics
- E-Commerce Dependability
- Failure Prevention
- Identity Management
- IPR of Security Technology
- Incident Response and Prevention
- Information Flow Control
- Internet Dependability
- Interoperability Aspects
- Intrusion Detection and Fraud Detection
- Legal Issues
- Mobile Security
- Network and Organizational Vulnerability Analysis
- Network Security
- Privacy-Enhancing Technologies Process based Security Models and Methods

Eduardo Fernández-Medina – Security in Software Engineering

9



**Multidisciplinaridad**

- RFID Security and Privacy
- Risk planning, Analysis & Awareness
- Safety Critical Systems
- Secure Enterprise Architectures
- Security Issues for Ubiquitous Systems
- Security and Privacy in E-Health
- Security and Trust Management in P2P and Grid applications
- Security and Privacy for Sensor Networks, Wireless/Mobile Devices and Applications
- Security and Usability
- Security as Quality of Service
- Security in Distributed Systems / Distributed Databases
- Security in Electronic Payments
- Security in Electronic Voting
- Software Engineering of Dependable Systems
- Software Security
- Standards, Guidelines and Certification

Eduardo Fernández-Medina – Security in Software Engineering

10




**Multidisciplinaridad**

- Survivability of Computing Systems
- Temporal Aspects of Dependability
- Threats and Attack Modelling
- Trusted Computing
- Tools for Dependable System Design and Evaluation
- Trust Models and Trust Management
- VOIP, Wireless Security
- Es **IMPOSIBLE** ser experto en seguridad informática
  - Sólo en una pequeña parte.

11

Eduardo Fernández-Medina – Security in Software Engineering



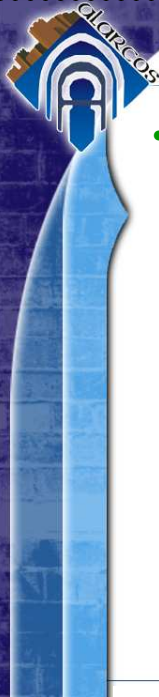
**Security in Software Engineering**

Content

- **Introduction**
- Secure IS Development – General Frameworks
- Security Requirements Engineering
- Security Architectural Patterns
- Model Driven Development
- Model Driven Security
- Secure Databases
- Secure Data Warehouses
- Secure Business Process Models
- Conclusions
- Events

12

Eduardo Fernández-Medina – Security in Software Engineering




**Security in Software Engineering**  
Introduction

- **Recommended Reading:**
  - **Book: Haralambos Mouratidis and Paolo Giorgini (2006). Integrating Security and Software Engineering. Advances and Future Visions. Idea Group Publishing.**
  - **Book: Haralambos Mouratidis. Software Engineering for Secure Systems: Industrial and Research Perspectives → Forthcoming in 2010.**

13

Eduardo Fernández-Medina – Security in Software Engineering



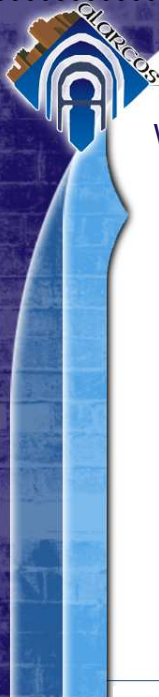
**Security in Software Engineering**  
Introduction

**What is Software Engineering (SE)?**

- Applying **engineering** to **software** (intuitive definition)
- Systematic approach for developing **software**.
- SE deals with the establishment of sound **engineering** principles and **methods** in order to economically obtain **software** that is reliable and works on real machines (Bauer, 1972).
- SE is the practical application of **scientific knowledge** in the design and construction of computer **programs** and the associated documentation required to develop, operate, and maintain them (Bohem, 1976).

14

Eduardo Fernández-Medina – Security in Software Engineering




**Security in Software Engineering**  
Introduction

### What is Software Engineering (SE)?

- SE is the study of the principles and **methodologies** for developing and maintaining **software** systems (Zelkowitz, 1978)
- SE is the application of a **systematic**, disciplined, quantifiable approach to the development, operation, and maintenance of **software** (IEEE, 1993).
- SE is a discipline of **engineering** which goal is the development of **software** systems with a reasonable cost (Sommerville, 2002)

15

Eduardo Fernández-Medina – Security in Software Engineering



**Security in Software Engineering**  
Introduction

### Main goal of SE:


- Providing methodologies, techniques, models, tools, etc. for the **correct development** of the **correct information system**.

### Typical processes in SE for developing information systems:

- Analysis – requirement definition
- Design – high level construction
- Construction – low level code generation
- Testing
- Maintenance
- Others: Planning, documentation, configuration management, quality assurance, and so on.

16

Eduardo Fernández-Medina – Security in Software Engineering



**Security in Software Engineering**  
Introduction

Traditional way of developing information systems:


- Complexity: Simple and medium systems
- Method: no method, no analysis, no design, informal requirement specification
- →Results: unpredictable. The developed systems is not what the client wants.
- →Cost: much more than the estimated cost
- →Time: much more than the estimated duration
- **Main problem:** Developers do not understand the details of the systems they develop.

Current and future way of developing information systems:

- Complexity: Complex and very complex systems
- Method: software engineering, methodologies, **special attention on requirements**
- →Results: The developed system is close to the client needs.
- →Cost: more or less predictable.
- →Time: more or less predictable.
- **Problem solved?** NO, we have improved the development process, but the system complexity have been increasing.

17

Eduardo Fernández-Medina – Security in Software Engineering

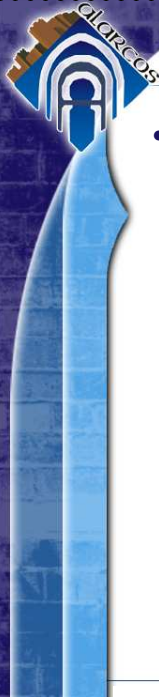


**Security in Software Engineering**  
Introduction

- Crucial (early **requirements**):
  - If we do not pay effort in knowing what the client needs, we will develop a system which does **not satisfy** the client needs.
  - We will realize that the system does not satisfy the client needs once the system has been developed, and therefore, once we have waste **too much** resources.
  - So, it is crucial to analyze, elicitate, specify and model information system **requirements**.
- What is a requirement?
  - (IEEE) System **requirements** specify a condition or capability needed by a user to solve a problem or achieve an objective, or that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.
    - Ex: The system prints invoices
    - Ex. The system generates a report with the sales distribution by district.

18

Eduardo Fernández-Medina – Security in Software Engineering




Security in Software Engineering  
Introduction

- However.....
  - These (**functional**) requirements are not the only requirements. These requirements represent the functionality of the system, but there are too many ways to provide this functionality.
  - Also important..... **Nonfunctional** requirements, which do not describe what the software will do, but **how** the software will do it.
    - software performance requirements
    - software external interface requirements
    - software design constraints
    - software quality attributes
    - software **security**
    - ...

19

Eduardo Fernández-Medina – Security in Software Engineering




Security in Software Engineering  
Introduction

- One of the most important AXIOMS of SE:
  - Once a **requirement** has been identified, its essence can be integrated into the system in the state the system is in this moment.
    - If a requirement is **early** identified, the system development will take into account it from the initial models, and the design will **perfectly support** it.
    - However, if a requirement is **lately** identified, the system will not be ready to accommodate the requirement, and we will need to integrate it modifying the existing components, adding new software **mistakes**, and probably **not** completely **respecting** that requirement.
      - Ex. Imagine we build a house without windows. If we realize about this **requirement** once the house has been built, we will need to broke the house walls, redistribute water and electricity infrastructure, paint again, etc. in order to integrate them.

20

Eduardo Fernández-Medina – Security in Software Engineering

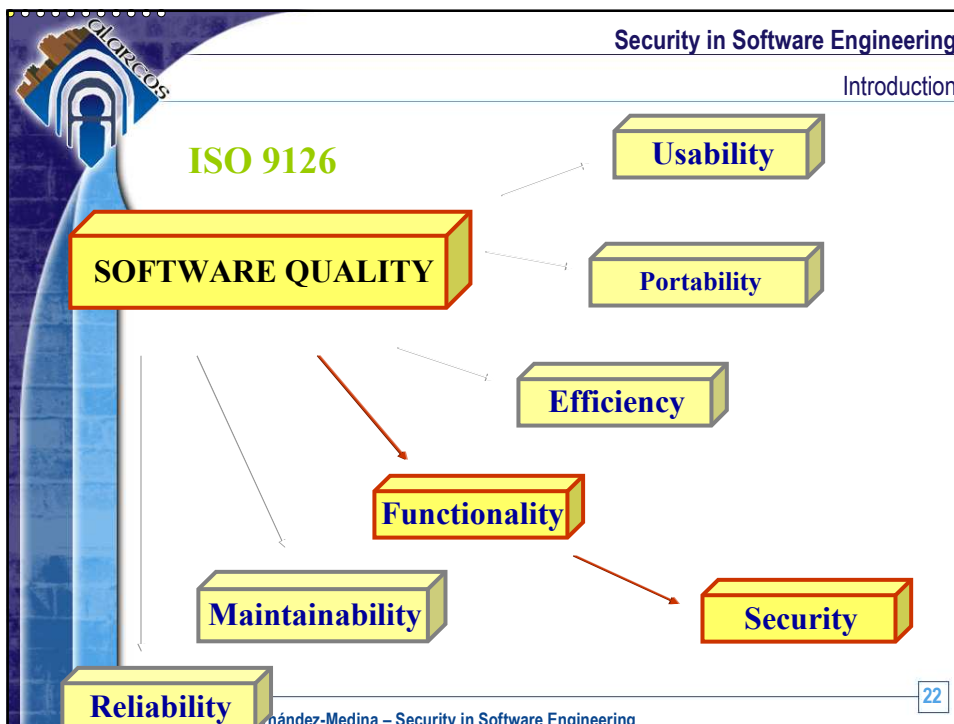
Security in Software Engineering  
Introduction

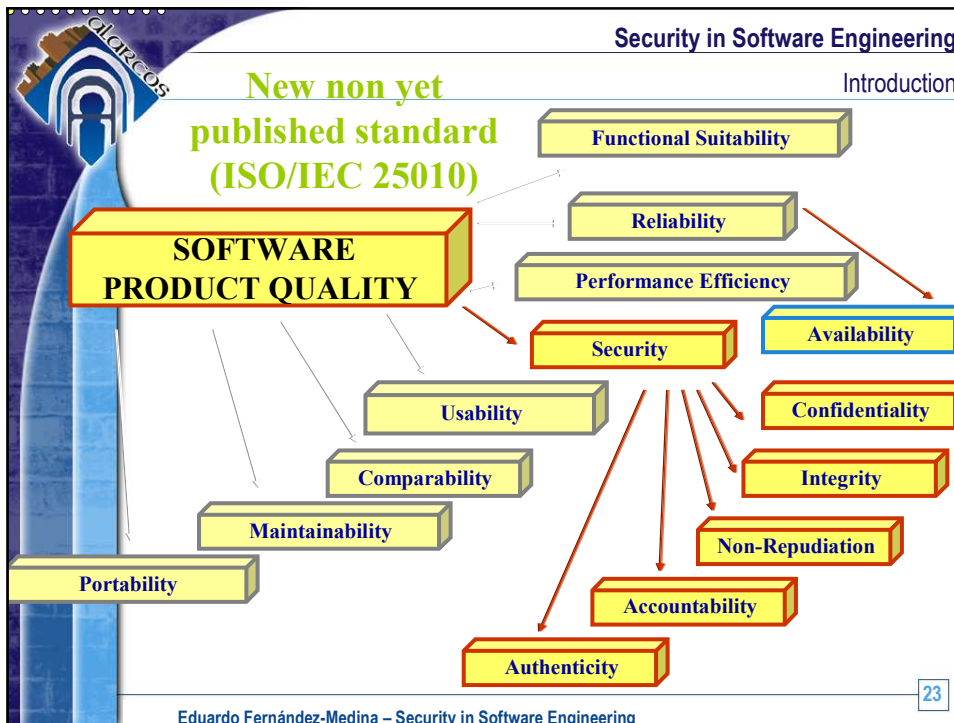


- Therefore....
  - We need to identify and specify software requirements as **soon** as possible, because we will develop more **robust** information systems, with much more **quality** and with much **less** resources.
  - But... which software requirements? Both functional and nonfunctional.
- But, what is software **quality**?

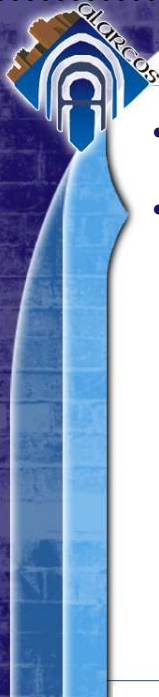
21

Eduardo Fernández-Medina – Security in Software Engineering





- Security in Software Engineering  
Introduction
- Security as a **characteristic of Software Product Quality**.
    - Security: the degree of protection of information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them.
      - **Confidentiality**: the degree of protection from unauthorized disclosure of data or information, whether accidental or deliberate.
      - **Integrity**: the degree to which a system or component prevents unauthorized access to, or modification of, computer programs or data.
      - **Non-repudiation**: the degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later.
      - **Accountability**: the degree to which the actions of an entity can be traced uniquely to the entity.
      - **Authenticity**: the degree to which the identity of a subject or resource can be proved to be the one claimed
      - **Availability (from reliability)**: the degree to which a system or component is operational and accessible when required for use.
- 24
- Eduardo Fernández-Medina – Security in Software Engineering




Security in Software Engineering  
Introduction

- Security is being much more **important** (considered as a characteristic of software quality).
- If we want to develop **quality software**, we need to consider security requirements.
  - **When?**
    - From the beginning of the development
  - **Why?**
    - Because in other case, if we develop our systems **without** taking into account security, we will be able to implement some technical security measures once the system have been developed, but the system **will not be** designed thinking in **security**.
      - Ex. Imagine we build a car, and then we realize we want our car to incorporate airbag as security measure. Can we do it?, yes, but it is difficult if the design of our car have not been prepared. However, we can design our car thinking in security as any other requirement, and our system will be robust.

25

Eduardo Fernández-Medina – Security in Software Engineering

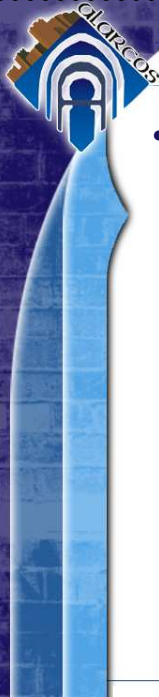


Security in Software Engineering  
Introduction

- However,
  - even though security is an important type of requirement, it has **usually** been considered superficially and the usual process is to include some standard security mechanisms **at the end** of the development.
  - traditional software engineers deal with security **after** the definition of the system, and after important **decisions** about the **design** of the system have been taken.
- On the opposite, security requirements must be **integrated** within the software development process, from its early stages, in order to create serious solutions with appropriate analysis and design **models**, which couple all requirements of the systems.

26

Eduardo Fernández-Medina – Security in Software Engineering




Security in Software Engineering  
Introduction

- This problem can be due by the reason of **software engineering** and **security engineering** have been traditionally **independent**:
  - On one hand, **software engineering** is focused on the systematic development of information systems, and do not consider security as an important issue. It recognize the importance of security as nonfunctional requirement, but software engineering techniques and methods do not incorporate security.
  - On the other hand, **security engineering** is focused on the definition of formal and theoretical methods (protocols, cryptographic algorithms, access control policies, information flow control, etc., etc., etc.), that usually are not aligned with information systems concepts and elements.

27

Eduardo Fernández-Medina – Security in Software Engineering



Security in Software Engineering  
Introduction

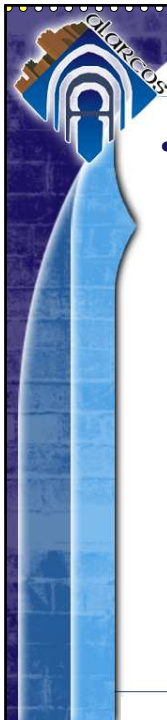
- I consider that “**Security in Software Engineering**” (also defined as “Secure Software Engineering” by authors such as Haris Mouratidis and Paolo Giorgini) is an **open research topic**, but in the last decade there have been a clear explosion of this research area, and the main contributions are related to the following aspects:
  - Integration of security into requirements engineering – security requirements engineering
  - Integration of security into software architectures – security architectural patterns
  - Integration of security into system models – modeling security together functionality
  - Integration of security into information system development processes

28

Eduardo Fernández-Medina – Security in Software Engineering

Security in Software Engineering	
	Content
<ul style="list-style-type: none"> <li>• Introduction</li> <li>• <b>Secure IS Development – General Frameworks</b></li> <li>• Security Requirements Engineering</li> <li>• Security Architectural Patterns</li> <li>• Model Driven Development</li> <li>• Model Driven Security</li> <li>• Secure Databases</li> <li>• Secure Data Warehouses</li> <li>• Secure Business Process Models</li> <li>• Conclusions</li> <li>• Events</li> </ul>	
29	
Eduardo Fernández-Medina – Security in Software Engineering	

Security in Software Engineering	
	Secure IS Development – General Frameworks
<ul style="list-style-type: none"> <li>• Recommended Reading: <ul style="list-style-type: none"> <li>▪ [JCR2] Villarroel, R., Fernández-Medina, E. and Piattini, M. (2005). Secure Information Systems Development – a survey and comparison. <i>Computers &amp; Security</i>. 24, 308-321.</li> <li>▪ [JCR3] Gutiérrez, C., Fernández-Medina, E., and Piattini, M. (2006). Towards a Process for Web Services Security. <i>Journal of Research and Practice in Information Technology</i>, 38 (1), 57-67.</li> <li>▪ [JCR11] Gutiérrez, C., Fernández-Medina, E., and Piattini, M. (2007). Web Services-based Security Requirement Elicitation. <i>IEICE Transactions on Information and Systems</i>, E90-D (9), 1374-1387</li> <li>▪ [JCR19] Gutiérrez, C., Rosado, D.G., and Fernández-Medina, E. (2009). The Practical Application of a Process for Eliciting and Designing Security in Web Service Systems. <i>Information and Software Technology</i>. In Press.</li> </ul> </li> </ul>	
30	
Eduardo Fernández-Medina – Security in Software Engineering	

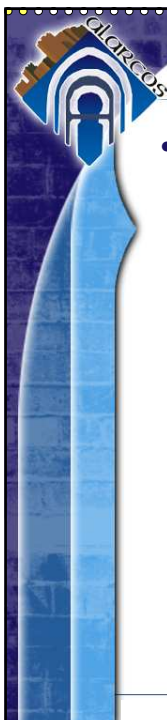


**Security in Software Engineering**  
Secure IS Development – General Frameworks

- So... we are realizing in the last decade that:
  - Security in information systems is **usually** considered once the information system **has been developed**.
  - This approach (Penetrate and Patch) has been proved to have **bad** results.
    - Solutions are mainly focused on providing **security defenses** (firewalls, routers, configuration servers, passwords, cryptography, etc.), instead of on integrating security decisions on the **software design**.
    - In simple **economic** terms, to find and eliminate mistakes in a software system when it is being developed is **cheaper** and **more effective** than to try to correct systems after having been finished.

31

Eduardo Fernández-Medina – Security in Software Engineering

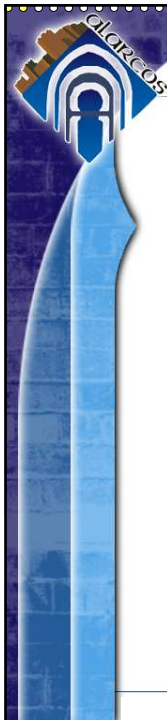


**Security in Software Engineering**  
Secure IS Development – General Frameworks

- Many approaches defining methods which integrate security “aspects” into the information systems development:
  - **MOMT**: Multilevel Object Modeling Technique:
    - An adaptation of the **OMT** methodology for developing secure databases.
  - **Business process-driven** framework for security engineering:
    - It is based on UML and integrates security requirements into a business process model of the system, such as non-repudiation, confidentiality, integrity, access control and authentication.
  - **UMLSec**: Secure Systems Development Methodology using UML:
    - This approach extends several **UML models** for specifying confidentiality and integrity requirements, specially oriented to multilevel security and mandatory access control.

32

Eduardo Fernández-Medina – Security in Software Engineering

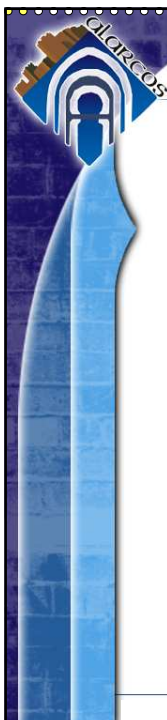


**Security in Software Engineering**  
Secure IS Development – General Frameworks

- **Secure Database Design Methodology:**
  - It is a methodology to design multilevel databases by integrating security (**confidentiality** constraints) into each one of the stages of the database life cycle.
- **CoSMo: Conceptual Security Modeling:**
  - It is a conceptual model that integrate security requirements, and then, trying to define security mechanisms that enforce these security requirements.
- **Methodology for Secure Software Design**
  - A methodology, composed by requirements, analysis, design and implementation stages, which tries to have traceability between these stages, and which is based on **RBAC**, security architectural **patterns** and the **OCL** language for specifying security constraints.
- **Secure Unified Process:**
  - Security is represented as a set of features that fortifies the application or service with safeguards and countermeasures for potential risk and vulnerabilities. Activities: Security Requirements, Security Architecture, Security Design, Security Implementation, White Box Testing, Black Box Testing, Monitoring and Security Auditing.

33

Eduardo Fernández-Medina – Security in Software Engineering

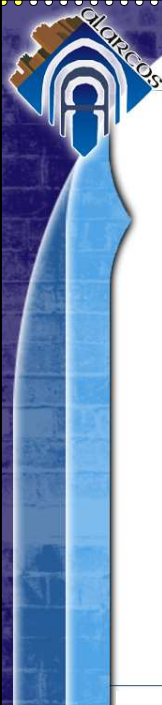


**Security in Software Engineering**  
Secure IS Development – General Frameworks

- **Secure Tropos.**
  - This is an agent oriented software engineering methodology which integrate security concepts (such as Security Constraint, Secure entity, Ownership, etc.) into their stages: Early Requirements Analysis, Late Requirements Analysis, Architectural Design and Detailed Design.
- **CLASP (Comprehensive, Lightweight Application Security Process):**
  - It is an activity-driven, role-based set of process components whose core contains formalized best practices for building security into your existing or new-start software development lifecycles in a structured, repeatable, and measurable way. That is to say, it offers a set of secure-oriented sub-processes that can be integrated into any development process.

34

Eduardo Fernández-Medina – Security in Software Engineering

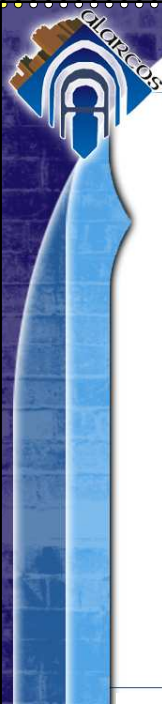


**Security in Software Engineering**  
Secure IS Development – General Frameworks

- **PWSSec: Process for Web Services Security**
  - This process facilitate the development of **WS-based security** systems so that, in each one of the traditional stages for the construction of this kind of systems, a complementary stage comprising security can be easily integrated.
  - The result will be a pattern-oriented security architecture formed by a set of coordinated **security mechanisms** that use the WS security standards to address the system security requirements.
  - It offers **risk-based** security engineering, a **pattern-oriented** security architecture and an **standard-centered** security design.

35

Eduardo Fernández-Medina – Security in Software Engineering

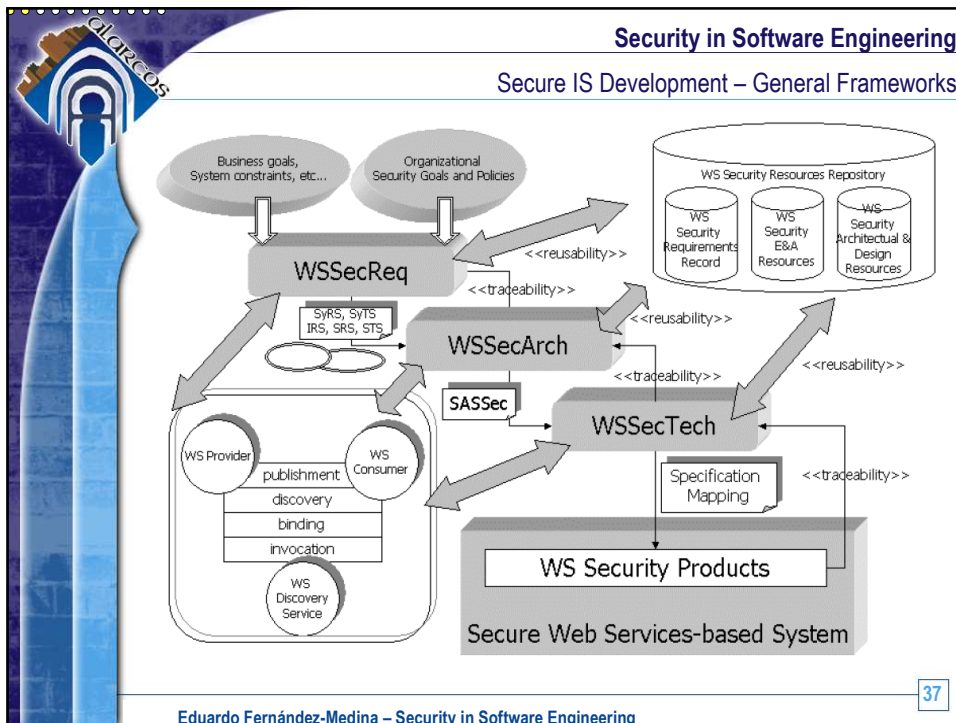


**Security in Software Engineering**  
Secure IS Development – General Frameworks

- **PWSSec: Process for Web Services Security**
  - Three subprocesses:
    - **WSSecReq: Web Services Security Requirements.**
      - » It identify the set of security **threats** over the system.
      - » It define threats trough **security artifacts**: threat trees,
      - » **Attack scenarios** are defined (misuse cases),
      - » **Risk** and attacks' impact are analyzed,
      - » Sound **countermeasures** are identified and modeled (security use cases),
      - » and finally **security requirements** are specified.
    - **WSSecArch: Web Services Security Architecture.**
      - » Allocate the security requirements specified in the WSSecReq subprocess into a WS-based security architecture
    - **WSSecTech: Web Services Security Technologies:**
      - » Identify the set of WS-based security standards that will implement the architectural security mechanisms identified in the WSSecArch subprocess.

36

Eduardo Fernández-Medina – Security in Software Engineering



### Security in Software Engineering

#### Specification Techniques

Table 1 Comparison using evaluation criteria for software specifications and specification techniques

Technique criterion	Specification criteria	Methodologies	Marks et al. (1996)	Vivas et al. (2003)	Jurjens (2002)	Fernández-Medina and Plattini (2003)	Liu et al. (2003)	Siponen et al. (2002)	Artelsmair et al. (2002)	Georg et al. (2002)	Fernández-Medina et al. (2004)	Priebe and Pernul (2001)	Fernández-Medina et al. (2004)
Expressive adequacy	Understandable	X	X	X	X	X	X	X	X	X	X	X	X
	Appropriate	(x)	X	(x)	X	(x)	(x)	X	X	X	X	X	X
	Minimal	X	X	X	X	(x)	(x)	X	X	X	X	X	X
Constructibility	-	X	X	X	X	(x)	(x)	(x)	(x)	(x)	(x)	X	X
	Complete	(x)	X	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	X
Level of formality	Unambiguous	X	(x)	X	X	X	(x)	(x)	X	(x)	X	X	X
	Consistent	X	(x)	X	X	X	(x)	(x)	X	(x)	X	X	X
	Complete	X	(x)	X	(x)	X	(x)	(x)	X	(x)	X	X	X
Formal foundation	Verifiable	X	(x)	X	X	X	(x)	(x)	X	(x)	X	X	X
	Validateable	X	(x)	X	X	X	(x)	(x)	X	(x)	X	X	X
	Unambiguous	X	(x)	X	X	X	X	X	X	X	X	X	X
Extent of applicability	Consistent	X	X	X	X	X	X	X	X	X	X	X	X
	Complete	X	(x)	X	(x)	X	X	X	X	X	X	X	X
	Validateable	(x)	X	(x)	X	(x)	X	X	X	X	X	X	X
Easy to use	-	X	X	X	X	(x)	X	(x)	(x)	(x)	X	X	X
	Help support	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)
	Integrated environment and tool support	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)
Specification organization	Understandable	X	(x)	X	X	X	(x)	(x)	(x)	(x)	(x)	(x)	(x)
	Modifiable	X	X	X	X	X	(x)	(x)	(x)	(x)	(x)	(x)	(x)
Support for maintainability	Modifiable	X	(x)	X	X	X	X	X	X	X	X	X	X
	Traceable	(x)	(x)	X	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)
Executable	Understandable	(x)	(x)	X	X	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)
	Unambiguous	(x)	(x)	X	X	X	(x)	(x)	(x)	(x)	(x)	(x)	(x)
	Consistent	(x)	(x)	X	X	X	(x)	(x)	(x)	(x)	(x)	(x)	(x)
	Complete	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)
	Verifiable	(x)	(x)	X	X	X	(x)	(x)	(x)	(x)	(x)	(x)	(x)
Tolerance for incompleteness	Verifiable	X	X	X	(x)	X	X	(x)	(x)	(x)	(x)	(x)	(x)
	Validateable	X	X	X	(x)	X	X	(x)	(x)	(x)	(x)	(x)	(x)
	Understandable	X	(x)	X	X	X	(x)	(x)	(x)	(x)	(x)	(x)	(x)
Flexibility and notational simplicity	Understandable	X	X	X	X	(x)	(x)	(x)	(x)	(x)	X	X	X
	Unambiguous	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)
Internal verification support	Complete	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)
	Consistent	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)
External validation support	Correct	X	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)
	Validateable	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)
Support for other development stages	Traceable	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)
Support for documentation generation	Understandable	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)	(x)

38

**Security in Software Engineering**  
Secure IS Development – General Frameworks

Table 2 Summary of the contributions, in security terms, made by each one of the analysed methodologies

	Modeling /development standard	Technologies	Access control type	Constraints specification	Case tool support
MOMT Vivas	OMT UML	Databases Information systems (only requirements, business process-driven)	MAC –	NO NO	NO YES (ConGolog, a concurrent logic programming based on the situation calculus)
UMLSec	UML patterns	Information systems	MAC (multilevel)		NO, but work towards this goal is being undertaken by giving translations from UML into CSP which allow us to use the model checker FDR2 to check security properties
Fernández-Medina and Plattini	UML, unified process	Databases	MAC, DAC, RBAC, Constraint-based	OSCL (OCL based)	YES (Rational Rose add-in)
Liu and Yu	Agent-oriented requirement modeling language (*)	Information systems (only requirements)	RBAC	A lightweight object modeling notation alloy	YES (alloy)
Siponen	–	Information systems, meta-methodology	–	NO	NO
CoSMo	UML	Information systems (only requirements)	–	–	NO
George et al.	UML, aspect-oriented patterns	Information systems (only design)	RBAC	They do not show the meta-model constraints in OCL, the constraints are expressed in template form	NO
Fernández	UML patterns	Information systems	Access matrix RBAC	He refers to OCL as a good solution	NO
ADAPTed UML	ADAPT UML	OLAP	RBAC	MDSCL (MDX based)	NO
Fernández-Medina et al.	UML	Data warehouses	MAC, DAC, RBAC, constraint-based	OSCL (OCL based)	NO

39

Eduardo Fernández-Medina – Security in Software Engineering

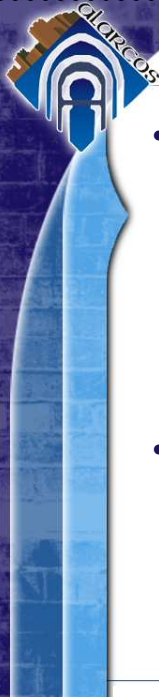
- Security in Software Engineering**  
Secure IS Development – General Frameworks
- These approaches are **very interesting**, and they contribute with important advances in the scientific community. However, they do **not offer complete and definitive** methods, and unfortunately, they do not reach the industry.
  - There are other recent approaches which are more **closed** of the current and **modern development** approaches, such as:
    - **Model Driven Security:**
      - Is a new approach of building secure information systems, in which designers specify **high-level** system models along with their security properties and use tools to **automatically generate** system architectures from the models, including security infrastructures. This proposal extends **Model Driven Architecture** with security models.
    - **SECTET: Framework for model driven security.**
      - Is a MDS based framework for B2B workflows. It applies **MDS** as the basis for many aspects, such as for specifying role and constraint based access control policies, for trust management, etc.
- 40
- Eduardo Fernández-Medina – Security in Software Engineering

Security in Software Engineering	
Content	
• Introduction	
• Secure IS Development – General Frameworks	
• <b>Security Requirements Engineering</b>	
• Security Architectural Patterns	
• Model Driven Development	
• Model Driven Security	
• Secure Databases	
• Secure Data Warehouses	
• Secure Business Process Models	
• Conclusions	
• Events	

Eduardo Fernández-Medina – Security in Software Engineering 41

Security in Software Engineering	
Security Requirements Engineering	
• Recommended Reading:	
▪ Mead, N. R., Hough, E. and Stehney, T. Security Quality Requirements Engineering (SQUARE) Methodology ( <a href="http://www.sei.cmu.edu/publications/documents/05.reports/05tr009.html">CMU/SEI-2005-TR-009</a> ). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005.	
• <a href="http://www.sei.cmu.edu/publications/documents/05.reports/05tr009.html">http://www.sei.cmu.edu/publications/documents/05.reports/05tr009.html</a>	
▪ [JCR6] Mellado, D., Fernández-Medina, E. and Piattini, M. (2007). A common criteria based security requirements engineering process for the development of secure information systems. <i>Computer Standards &amp; Interfaces</i> . 29 (2007), 244-253.	
▪ [JCR15] Mellado, D., Fernández-Medina, E. and Piattini, M. (2008). Towards security requirements management for software product lines: A security domain requirement engineering process. <i>Computer Standards &amp; Interfaces</i> . 30 (2008), 361-371.	
▪ [CIN42] Mellado, D., Fernández-Medina, E. and Piattini, M. (2006). A Comparative study of proposals for establishing security requirements for the development of secure information systems. LCNS 3982. 1044-1053.	

Eduardo Fernández-Medina – Security in Software Engineering 42




**Security in Software Engineering**  
Security Requirements Engineering

- **Requirements engineering** is the branch of **software engineering** concerned with the real world **goals** for, **functions** of, and **constraints** on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families (Nuseibeh and Easterbrook).
- **Security Requirements Engineering** is the branch of **requirements engineering** concerned with **security requirements elicitation, analysis, modeling and specification.**

43

Eduardo Fernández-Medina – Security in Software Engineering

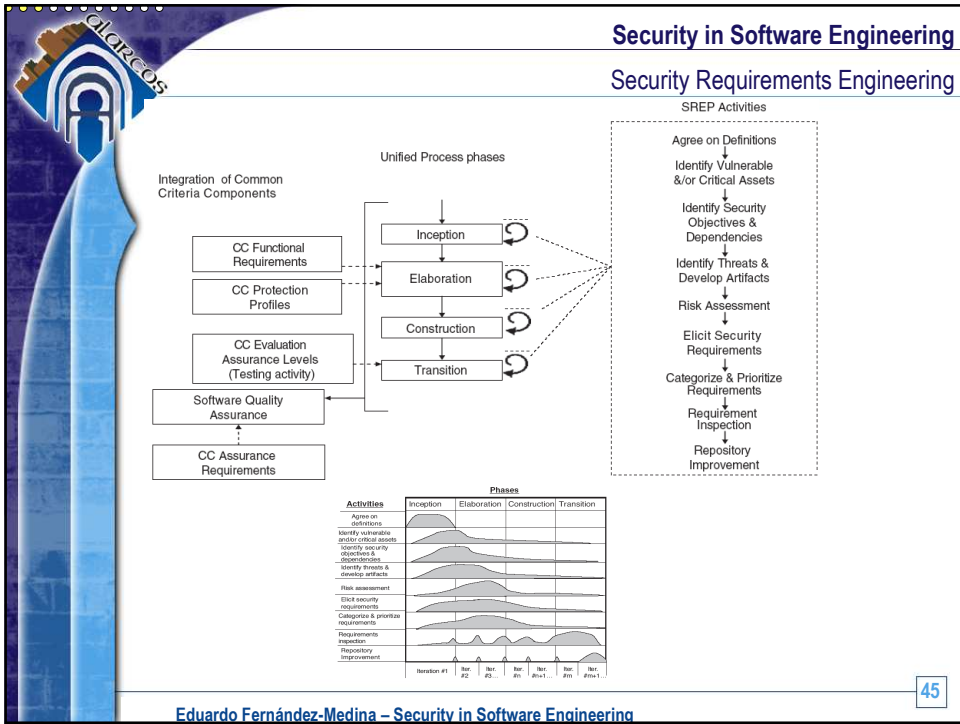


**Security in Software Engineering**  
Security Requirements Engineering

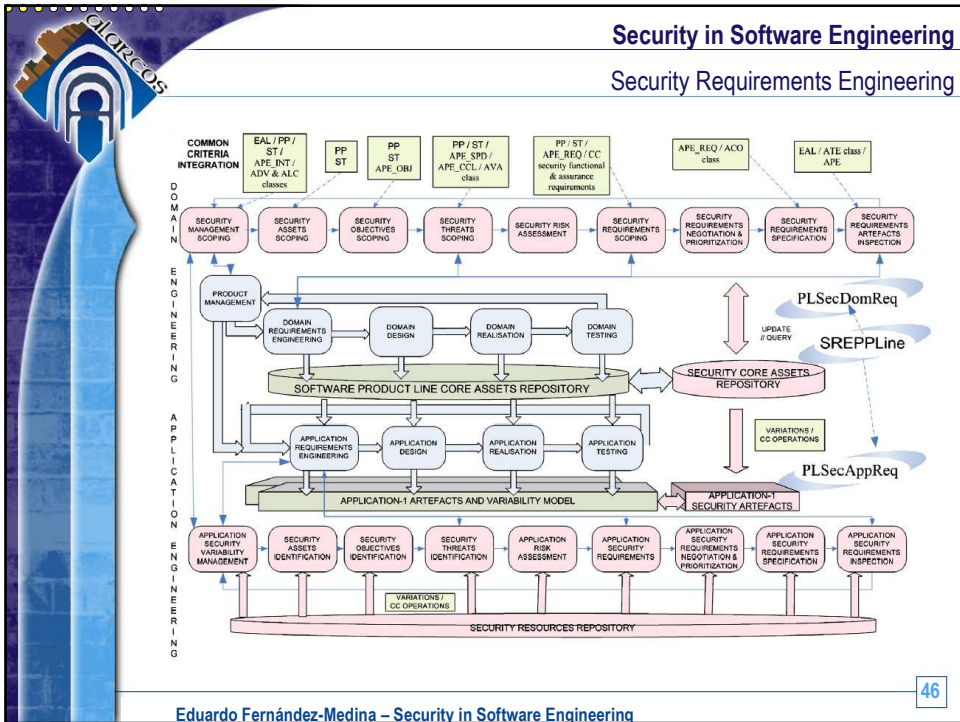
- Typical **activities** which are included in some of the most important approaches of security requirements engineering:
  - Agreement on **definitions** between stakeholders and engineering team
    - Example (SQUARE):
      - Access Control: set of policies that govern which users may be granted access to which resources
      - Access Control: The software elements in the system that actually implement this functionality.
  - **Identify and prioritize Security Goals**
  - Identify critical/vulnerable **assets**
  - **Modeling** requirements / Developing artifacts
  - Risk assessment
  - **Elicit** security requirements
  - Categorize and **prioritize** requirements
  - Requirements **inspection**

44

Eduardo Fernández-Medina – Security in Software Engineering



Eduardo Fernández-Medina – Security in Software Engineering



Eduardo Fernández-Medina – Security in Software Engineering

**Security in Software Engineering**  
**Security Requirements Engineering**

Technical criterion	Internal verification support						External validation support		Support for docum. generation	Standards integration			Requirem. reuse		Support for other development stages		Help support	Easy to use						
	Correct	Unambiguous	Modifiable	Validatable	Complete	Consistent	Traceable	Other by inspection	Correct	Validatable	Understandable	Understandable	Consistent	Verifiable	Complete	Consistent	Modifiable	Appropriate	Complete	Traceable	Modifiable	-----	-----	
Basin et al	*	*	*	P	P	*	*	X	X	X	*	X	X	X	X	X	X	X	P	P	*	*	*	
Bresciani et al. and Giorgini et al. and Masetti et al	*	*	*	*	*	*	*	*	*	*	*	P	P	P	P	*	*	*	*	*	*	*	*	
Firesmith	*	*	*	*	P	*	*	*	*	X	*	X	X	X	X	*	*	P	P	*	P	*	*	
Hussein and Zulkernine	*	*	*	*	P	X	X	P	X	*	X	X	X	X	X	X	X	P	P	X	*	*	P	
Jenney	*	*	*	*	*	*	*	*	*	*	*	X	X	X	X	X	X	X	P	P	X	*	*	
J. Lee, et al	*	*	*	*	*	*	*	X	*	*	*	*	*	*	*	X	X	X	P	*	*	*	*	P
S.-W. Lee et al.	*	*	*	*	*	*	*	X	X	X	*	*	*	*	*	*	*	P	*	*	*	X	*	P
Mead and Stehney	*	*	*	*	*	*	*	*	*	*	P	*	*	*	P	X	X	X	*	*	*	*	*	P
Mellado et al.	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	P	*	*
Moffett and Nuseibeh	*	*	*	*	*	*	*	P	*	*	*	X	X	X	X	X	X	X	*	*	*	*	*	*
Morimoto, et al	*	*	*	P	*	*	*	X	X	X	P	*	*	*	P	*	*	*	X	X	X	*	*	*
Myagmar et al.	P	P	P	P	P	P	P	X	X	X	P	X	X	X	X	X	X	X	X	X	X	*	*	P
Peters	P	P	P	P	P	P	X	X	X	*	X	X	X	X	X	X	X	P	P	P	*	*	*	
Popp et al. And Jürjens	*	*	*	*	P	*	*	*	*	X	*	X	X	X	X	X	X	X	P	P	*	*	*	*
Shin and Gomas	*	*	*	*	*	*	*	P	X	X	X	X	X	X	*	*	X	P	P	*	X	*	P	
Sindre and Opdahl. And Sindre et al.	*	*	*	*	*	P	*	*	*	X	P	X	X	X	X	*	*	P	*	*	P	*	*	*
Toval et al	*	*	*	*	*	*	*	*	*	P	*	*	*	X	P	*	*	*	P	*	*	*	*	*
Tsoumas and Grizalis	*	*	*	*	*	*	*	*	*	P	P	*	*	*	*	*	*	*	P	*	*	P	*	P
Viega	*	*	*	*	*	*	*	*	*	P	*	X	X	X	X	X	X	X	P	*	P	*	*	*
Yu	*	*	*	*	*	*	P	X	X	X	*	X	X	X	X	X	X	X	P	*	*	*	*	*
Zuccato	*	*	*	*	*	*	*	*	*	*	*	*	*	*	P	X	X	X	*	P	*	*	*	P

47

Eduardo Fernández-Medina – Security in Software Engineering

- Security in Software Engineering**  
**Security Requirements Engineering**
- These are examples of security requirements engineering processes. But one of the most important elements of this discipline are security **artifacts**, such as:
    - Misuse cases
    - Security use cases
    - UMLSec use cases
    - UMLSec clases
    - Attack trees diagram
    - Etc.
- 48
- Eduardo Fernández-Medina – Security in Software Engineering

Security in Software Engineering  
Security Requirements Engineering

- Misuse cases (Sindre and Opdahl)

Use case with hostile intention

The diagram illustrates the following relationships:

- Customer Actor:**
  - Includes: Browse catalog, Register customer, Order goods, Change password, Log on.
  - Extends: Register customer (with Block repeated registrations), Order goods (with Enforce password regime).
- Operator Actor:**
  - Includes: Log on.
  - Extends: Change password (with Enforce password regime).
- Security Use Cases (Black):**
  - Flood system:** Prevents Block repeated registrations; Detected by Crook.
  - Steal card info:** Detected by Crook; Includes Tap communication.
  - Obtain passwd:** Detected by Crook; Includes Tap communication.
  - Encrypt message:** Prevents Tap communication.
  - Monitor system:** Includes Log on and Obtain passwd.
- Other Use Cases (White):**
  - Block repeated registrations (Prevents Flood system)
  - Enforce password regime (Prevents Obtain passwd)
  - Log on (Includes Monitor system)

49

Eduardo Fernández-Medina – Security in Software Engineering

Security in Software Engineering  
Security Requirements Engineering

- Security Use Case (Firesmith)

The diagram illustrates the following relationships:

- Customer Actor:**
  - Includes: Deposit Funds, Withdraw Funds, Transfer Funds, Query Balance, Manage Accounts.
- Security Use Cases (White):**
  - Control Access (Security)
  - Ensure Privacy (Security)
  - Ensure Integrity (Security)
  - Ensure Nonrepudiation (Security)
- Misuse Cases (Black):**
  - Spoof User (Misuse)
  - Invade Privacy (Misuse)
  - Perpetrate Fraud (Misuse)
- Actors:**
  - Cracker:** Spoof User (Misuse)
  - Thief:** Invade Privacy (Misuse), Perpetrate Fraud (Misuse)
- Legend:**
  - White box: Security Use Case
  - Black box: Misuse Case
  - Stick figure: Misuser

50

Eduardo Fernández-Medina – Security in Software Engineering

Security in Software Engineering  
Security Requirements Engineering

- UMLSec models (Jürjens)

The slide displays two UMLSec models. The first, titled "Sales application", is annotated with the stereotype «fair exchange». It shows two actors: "Customer" and "Business". The Customer has a use case "buys goods", and the Business has a use case "sells goods". The second model, titled "Remote access", is annotated with «secure links». It shows two systems: "client machine" and "server machine". The client machine contains "client apps" and "browser". The server machine contains "web server" and "access control". Interactions include "get password" from the client to the server, "Internet" as a link, and "call" from the client to the server.

51

Eduardo Fernández-Medina – Security in Software Engineering


Security in Software Engineering  
Security Requirements Engineering

- Derived models. Ej.:

The diagram illustrates a complex set of derived models. It features several use cases and actors. Actors include "Journalist" (GridActor), "MobileUC" (MobileUC), "Authentication server" (GridActor), "Attacker" (MisActor), and "Unauthorized access" (MisuseCase). Use cases include "Login", "Authenticate", "Authorize Access", "Request", "Ensure Confidentiality", "Protect message", "Ensure Integrity", and "Alteration info". The diagram is heavily annotated with stereotypes: «threaten» (dashed arrows), «mitigate» (dashed arrows with a shield icon), «protect» (dashed arrows with a shield icon), and «permit» (dashed arrows with a shield icon). It also shows «include» relationships between use cases.

52

Eduardo Fernández-Medina – Security in Software Engineering




**Security in Software Engineering**  
Content

- Introduction
- Secure IS Development – General Frameworks
- Security Requirements Engineering
- **Security Architectural Patterns**
- Model Driven Development
- Model Driven Security
- Secure Databases
- Secure Data Warehouses
- Secure Business Process Models
- Conclusions
- Events

53

Eduardo Fernández-Medina – Security in Software Engineering




**Security in Software Engineering**  
Security Architectural Patterns

- **Recommended Reading:**
  - Schumacher, M., Fernandez-Buglioni, E., Hyberston, D., Buschmann, F., and Sommerlad, P. (2006). *Security Patterns, Integrating Security and Systems Engineering*, John Wiley & Song.
  - Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. and Stal, M. (1996), *Pattern-Oriented Software Architecture: A System of Patterns*, John Wiley & Sons.
  - [JCR9] Rosado, D.G., Gutiérrez, C., Fernández-Medina, E. and Piattini, M. (2006). Security Patterns and Requirements for Internet-based Applications. *Internet Research*. 16 (5), 519-536.
  - <http://www.securitypatterns.org/>

54

Eduardo Fernández-Medina – Security in Software Engineering

Security in Software Engineering  
Security Architectural Patterns




- A **Pattern** is a **solution** to a problem that arises within a specific context.
- (Buschmann et al., 1996) A **pattern** for software architecture describes a particular **recurring design** problem that arises in specific design context, and presents a **well-proven generic solution** for it. The solution consists of a set of interacting roles that can be arranged to form multiple concrete design structures, as well as a process for creating any particular structure.
  - They can be **reused** for similar situations.
  - They can help in the **design stage**.
  - They can help us to go from **requirements to implementation**.

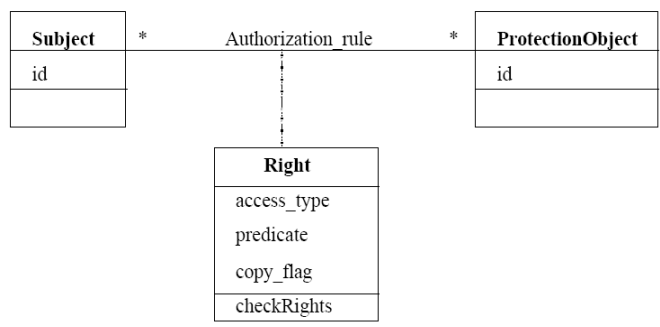
55

Eduardo Fernández-Medina – Security in Software Engineering

Security in Software Engineering  
Security Architectural Patterns



- Authorization Pattern



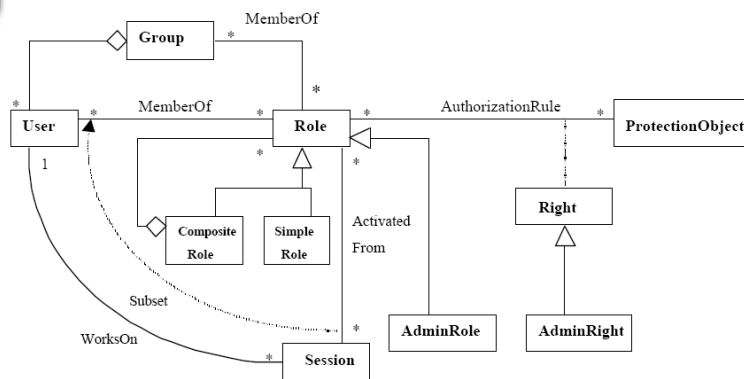
```

classDiagram
    class Subject {
        id
    }
    class ProtectionObject {
        id
    }
    class Right {
        access_type
        predicate
        copy_flag
        checkRights()
    }
    Subject "*" -- "*" ProtectionObject : Authorization_rule
    Right ..|> Authorization_rule
  
```

56

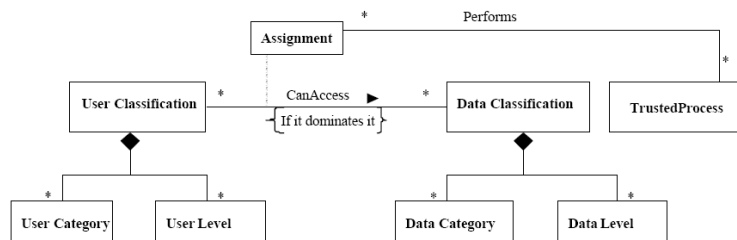
Eduardo Fernández-Medina – Security in Software Engineering

• RBAC Pattern

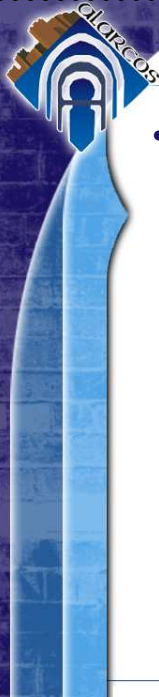


57

• MAC Pattern



58




**Security in Software Engineering**  
Security Architectural Patterns

- Many other type of patterns
  - Identification & Authentication
  - Access Control
  - Firewall Architecture Patterns
  - Accounting
  - Cryptographic Key Management
  - .....

59

Eduardo Fernández-Medina – Security in Software Engineering

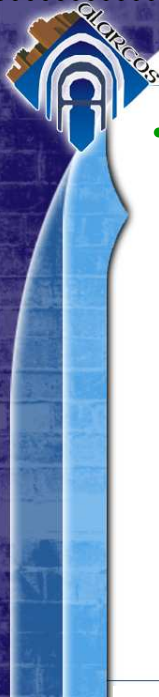


**Security in Software Engineering**  
Content

- Introduction
- Secure IS Development – General Frameworks
- Security Requirements Engineering
- Security Architectural Patterns
- **Model Driven Development**
- Model Driven Security
- Secure Databases
- Secure Data Warehouses
- Secure Business Process Models
- Conclusions
- Events

60

Eduardo Fernández-Medina – Security in Software Engineering




**Security in Software Engineering**  
Model Driven Development

- **Recommended Reading:**
  - J. Bézivin, In Search of a Basic Principle for Model Driven Engineering, Upgrade 5 (2) (2004) 21-24.
  - S. Mellor, K. Scott, A. Uhl, D. Weise, MDA Distilled: Principles of Model-Driven Architecture, Addison Wesley, 2004.
  - OMG, Model Driven Architecture Guide Version 1.0.1, 2003.
  - [JCR editorial 3] Fernández-Medina, E., Jurjens, J., Trujillo, J., and Jajodia, S. (2008). Model Driven Development for Secure Information Systems. *Information and Software Technology*. 51 (2009) 809-814.
    - Articles from the Special Issue on Model Driven Development for Secure Information Systems. *Information and Software Technology*.

61

Eduardo Fernández-Medina – Security in Software Engineering

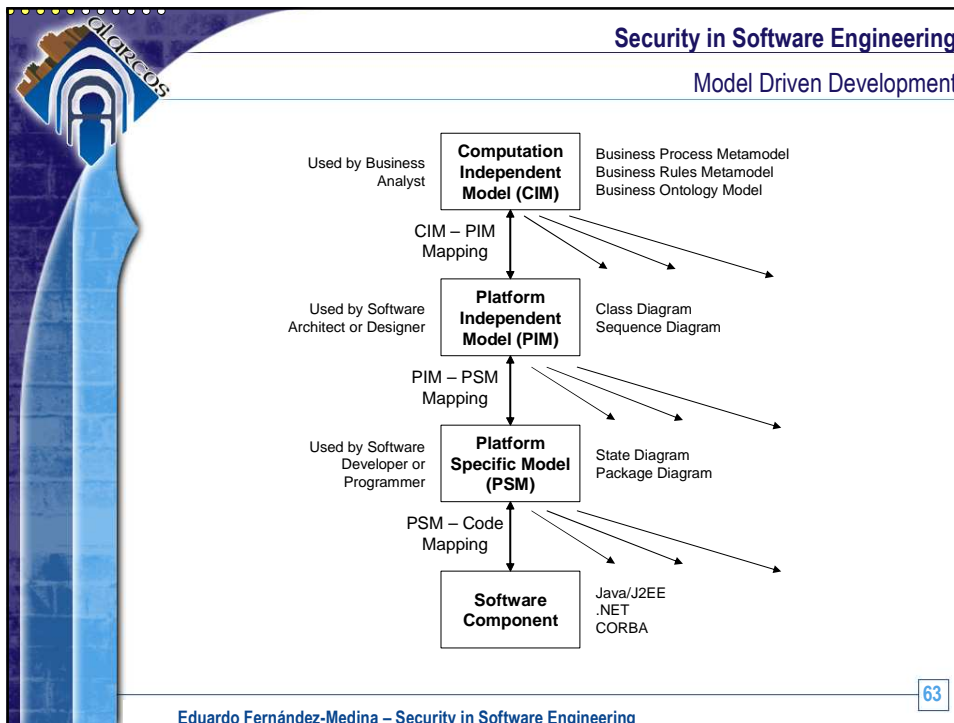


**Security in Software Engineering**  
Model Driven Development


- OO → Everything is an **object**
- MDD → Everything is a **model**
  - The goal of this important change is to attempt to solve (or at least improve) the historic **problems** of time, cost and quality in **software** development.
- MDE is the software engineering discipline which considers **models** as the most important element for software development, and for the maintenance and evolution of **software**, through model **transformation**.
- This discipline offers not only **independence** between models but also clearly separates the **business complexity** from the **implementation** details, by defining several software models at different **abstraction levels**.

62

Eduardo Fernández-Medina – Security in Software Engineering



- Security in Software Engineering  
Model Driven Development
- This architecture proposes **not only** a set of **models** that represent the system at different abstraction levels, but also a software development **life cycle** with which to:
    - i) capture **requirements** in a CIM,
    - ii) **create** a PIM (it is sometimes possible for part of the PIM to be obtained from the CIM),
    - iii) **transform** the PIM into one or more PSMs, adding platform-specific rules and code that the transformation did not provide;
    - iv) **transform** the PSM into **code**, and
    - v) **deploy** the **system** in a specific environment.
- 64
- Eduardo Fernández-Medina – Security in Software Engineering




**Security in Software Engineering**  
Model Driven Development

- Important: **Metamodels**, which define the elements of the models.
- Important: **Transformations**, which allows to the automatic or semi-automatic generation of low level models
  - → more importance of high level models (requirements)
- **Standard** MDD paradigm: OMG Model Driven Architecture – **MDA**
- Standard MDA **models**: OMG Unified Modeling Language - **UML**
- Standard **metamodel** definition: OMG Meta Object Facility – **MOF**
- Standard **transformation** language: Query / View / Transformation - **QVT**

65

Eduardo Fernández-Medina – Security in Software Engineering



**Security in Software Engineering**  
Model Driven Development

- This development paradigm is (more or less) close to the “**automatic development**”, where we specify the requirements of our system, and the implementation in our desired platform is automatic generated.
- However the process is **not so easy** and not so automatic.
- Some **security aspects** can be integrated into the abstract models, and then, these security aspects also can be **transformed** into more concrete models.
- See [**JCR editorial 3**] and the papers of the Special Issue.

66

Eduardo Fernández-Medina – Security in Software Engineering

Security in Software Engineering	
Content	
• Introduction	
• Secure IS Development – General Frameworks	
• Security Requirements Engineering	
• Security Architectural Patterns	
• Model Driven Development	
• <b>Model Driven Security</b>	
• Secure Databases	
• Secure Data Warehouses	
• Secure Business Process Models	
• Conclusions	
• Events	

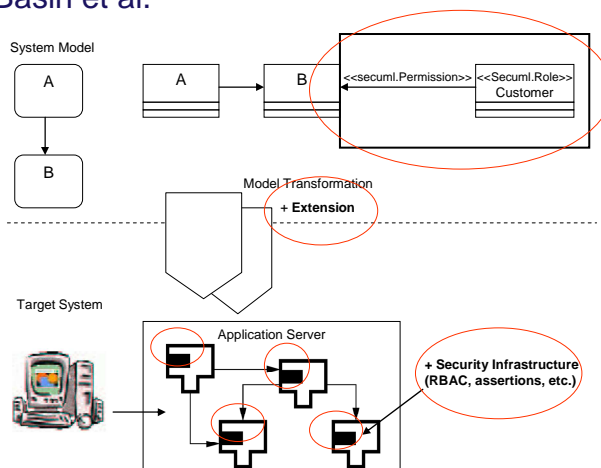
Eduardo Fernández-Medina – Security in Software Engineering 67

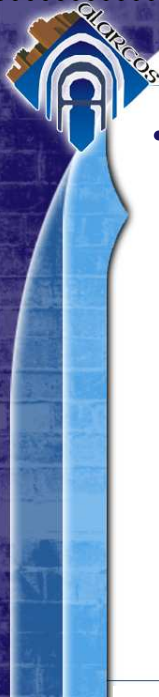
Security in Software Engineering	
Model Driven Security	
• <b>Recommended Reading:</b>	
▪ D. Basin, J. Doser, T. Lodderstedt, Model Driven Security for Process-oriented Systems, in: Proceedings of the ACM Symposium on Access Control Models and Technologies, Como, Italy, 2003, ACM Press, pp. 100-109.	
▪ D. Basin, J. Doser, T. Lodderstedt, Model Driven Security: from UML Models to Access Control Infrastructures, ACM Transactions on Software Engineering and Methodology 15 (1) (2006) 39-91.	
▪ <b>[JCR editorial 3]</b> Fernández-Medina, E., Jurjens, J., Trujillo, J., and Jajodia, S. (2008). Model Driven Development for Secure Information Systems. <i>Information and Software Technology</i> . 51 (2009) 809-814.	
• Articles from the Special Issue on Model Driven Development for Secure Information Systems. <i>Information and Software Technology</i> .	

Eduardo Fernández-Medina – Security in Software Engineering 68

- **MDS** is a new approach for building secure information systems, in which designers specify high-level system **models** along with their **security properties** and use tools to automatically **generate** system architectures from the models, including **security infrastructures**.
- MDS extends MDA in three aspects:
  - i) the system **models** are **enriched** with security aspects into the development process,
  - ii) **transformations** are **extended** to ensure that these security details are also transformed, and
  - iii) the **system** is obtained, including the security properties and the corresponding **security mechanisms**.

- Basin et al.






**Security in Software Engineering**  
Model Driven Security

- The general idea of MDA and MDS **opens** a new set of **research** possibilities based on the incorporation of security aspects into models, and providing transformations (both direct and reverse) of models.
  - SECTET (Breu et al.)
  - UMLSec (Jurgens et al.)
  - Databases and Data Warehouses
  - Business processes
  - Etc.

71

Eduardo Fernández-Medina – Security in Software Engineering

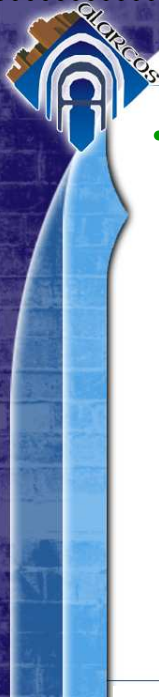


**Security in Software Engineering**  
Content

- Introduction
- Secure IS Development – General Frameworks
- Security Requirements Engineering
- Security Architectural Patterns
- Model Driven Development
- Model Driven Security
- **Secure Databases**
- Secure Data Warehouses
- Secure Business Process Models
- Conclusions
- Events

72

Eduardo Fernández-Medina – Security in Software Engineering




Security in Software Engineering

Secure Databases

- Recommended Reading:
  - [JCR 1] Fernández-Medina, E., and Mario Piattini (2005). Designing Secure Databases. *Information and Software Technology*, 47 (7), 463-477
  - [JCR 8] Vela, B., Fernández-Medina, E., Marcos, E., and Piattini, M. (2006). Model Driven Development of Secure XML Databases. *ACM Sigmod Record*, 35 (3), 22-27

Eduardo Fernández-Medina – Security in Software Engineering

73



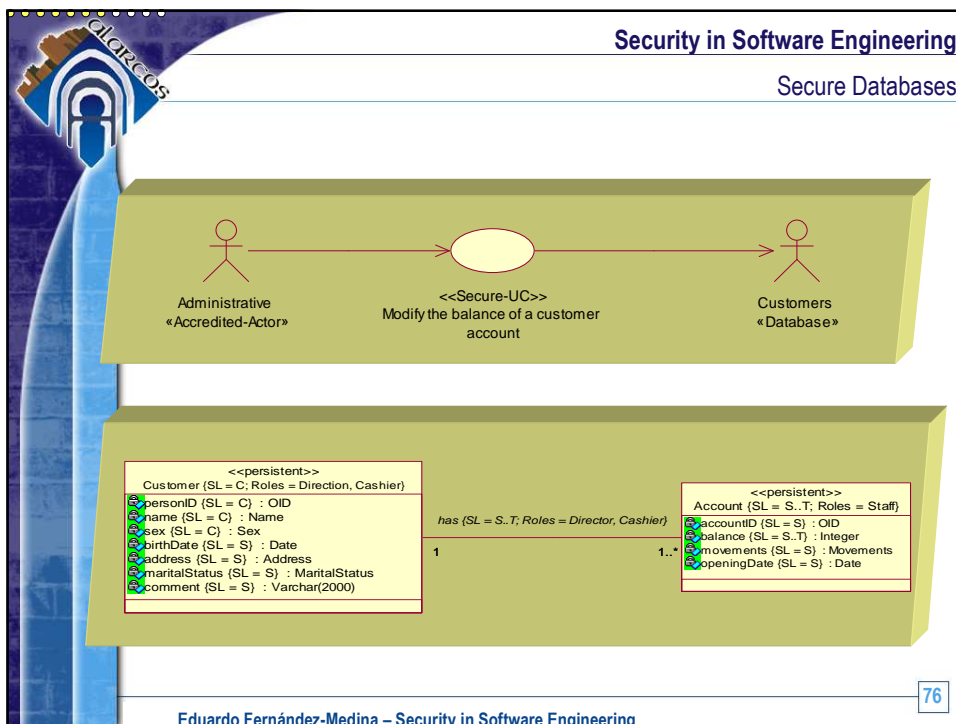
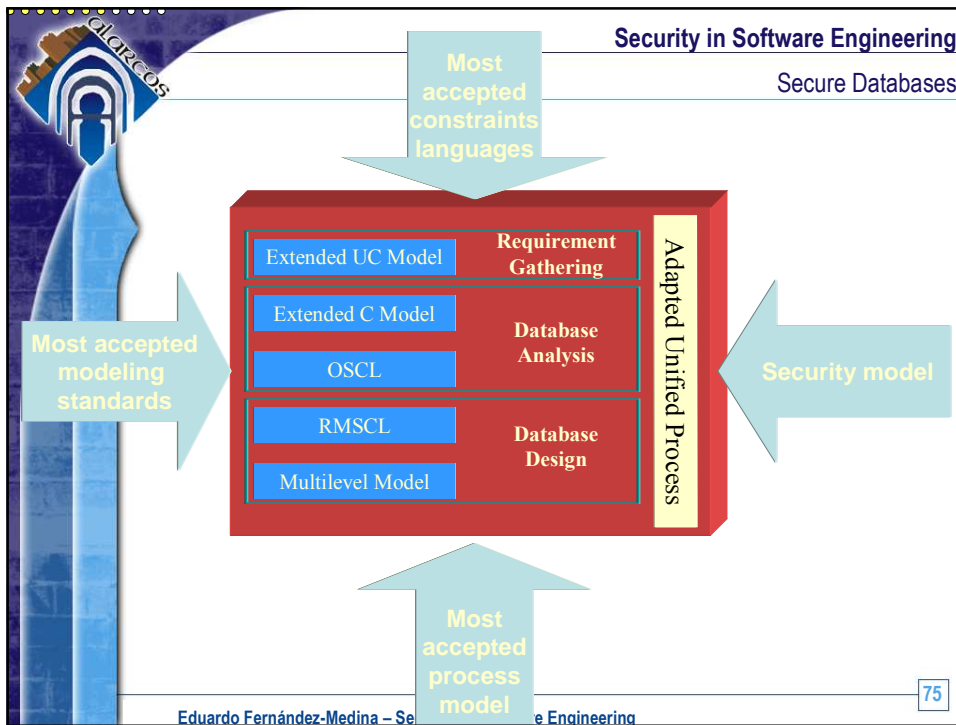
Security in Software Engineering

Secure Databases

- We developed an approach for the **systematic** development of **secure databases**, not completely MDA compliant, but with some initial related ideas:
  - Security aspects are **integrated** into the database **models**
  - Model **transformations** are manually performed
  - Different **paths** can be considered, but at the end of the architecture

Eduardo Fernández-Medina – Security in Software Engineering

74



Security in Software Engineering  
Secure Databases

Satellite\_image (SL = U .. TS)

- id : Integer
- image : RGB
- Accuracy : Integer
- Purpose : String
- Frequency : Real
- Compression\_Format : String

context Satellite\_image

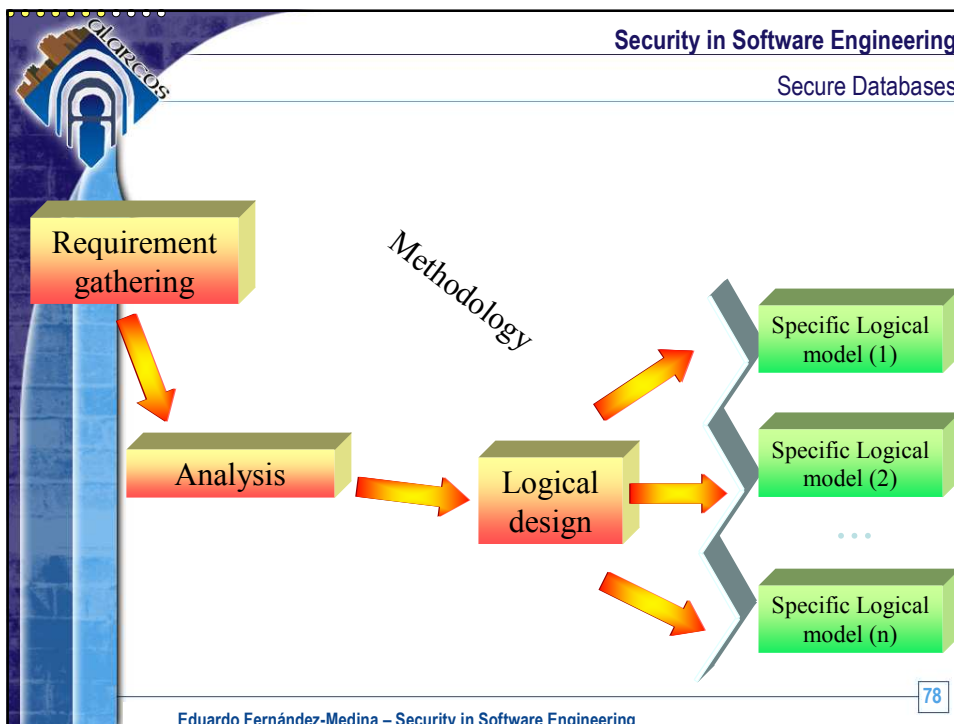
inv:

```
self.SL = (if self.Accuracy >= 90 then U else
(if self.Accuracy >= 10 then C else (if
self.Accuracy >= 2 then S else TS endif)
endif) endif)
```

**context Satellite\_Image inv:**

```
self.SL = (if self.Purpose = "Military"
then TS
else (if self.Purpose = "Spying"
then (if self.Accuracy >= 10
then S
else TS
endif)
else (if self.Purpose = "Maps"
then (if self.Accuracy >= 90
then U
else (if self.Accuracy >= 10
then C
else (if self.Accuracy >= 2
then S
else TS
endif)
endif)
endif)
endif)
endif)
endif)
```

Eduardo Fernández-Medina – Security in Software Engineering



Security in Software Engineering  
Secure Databases

**Activities**

- Requirements Gathering
  - Gathering initial requirements
  - Creating the business model and the system glossary
  - Looking for actors
  - Looking for use cases
  - Looking for permanent elements
  - Describing use cases
  - Analyzing security in actors and in use cases
  - Structuring the use case model
  - Defining priorities in use cases
  - Looking for relationships between use cases
  - Reviewing use cases

79

Eduardo Fernández-Medina – Security in Software Engineering

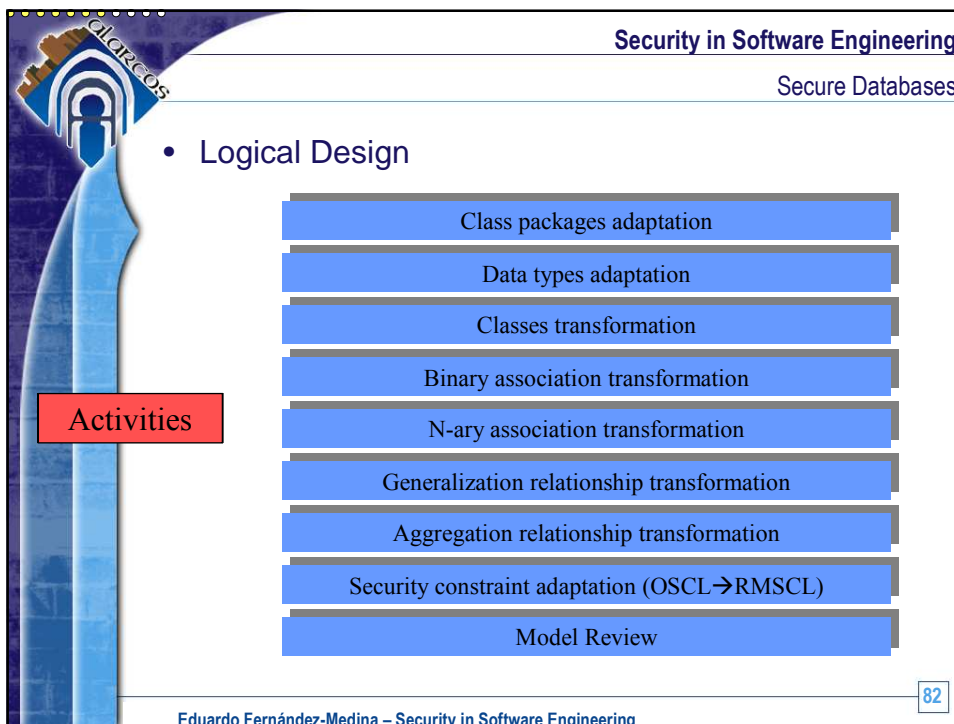
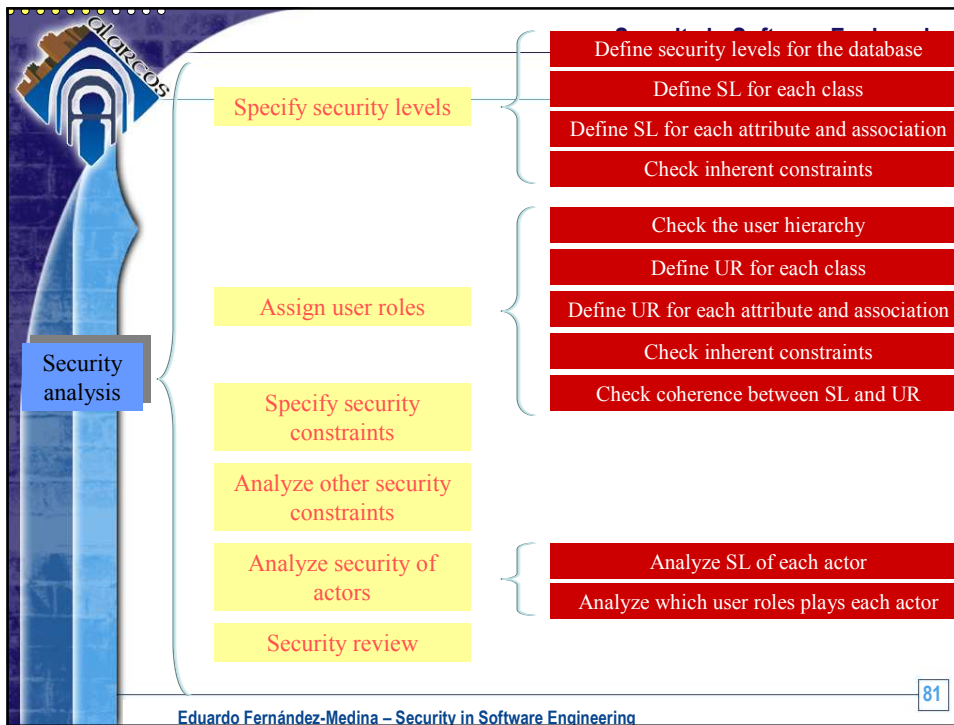
Security in Software Engineering  
Secure Databases

**Activities**

- Analysis
  - Architecture analysis
    - Identify analysis packages
    - Identify evident entity classes
    - Identify relationships
  - Use case analysis
    - Identify analysis classes of a UC
    - Classes review
  - Classes analysis
    - Identify attributes
    - Identify relationships
    - Attributes and associations review
  - Security analysis
  - Package analysis

80

Eduardo Fernández-Medina – Security in Software Engineering

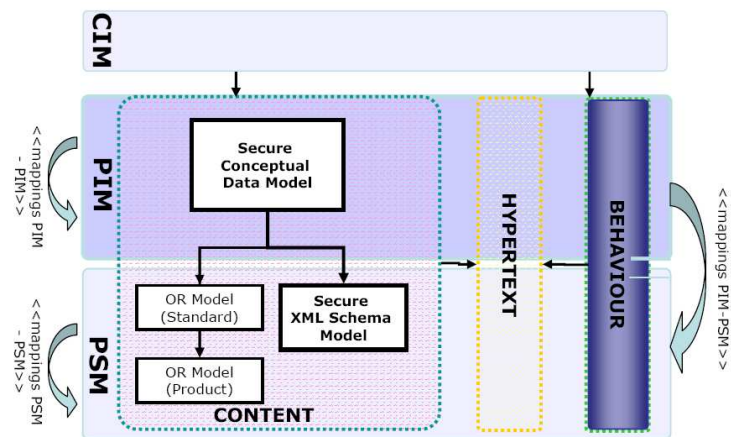


- Specific Logical Design (Oracle Label Security)

Activities

- Define database structure
- Create security policy and default options
- Definition of sensibility values for the security policy
- Definition of authorized users and assign them both accreditation information and the privileges if necessary
- Define labeling functions in order to assign sensibility information to the tables
- Define labeling functions and predicates in order to implement the security constraints
- Specify operations and check their security

- XML databases



Security in Software Engineering	
	Content
<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Secure IS Development – General Frameworks</li> <li>• Security Requirements Engineering</li> <li>• Security Architectural Patterns</li> <li>• Model Driven Development</li> <li>• Model Driven Security</li> <li>• Secure Databases</li> <li>• <b>Secure Data Warehouses</b></li> <li>• Secure Business Process Models</li> <li>• Conclusions</li> <li>• Events</li> </ul>	
85	
Eduardo Fernández-Medina – Security in Software Engineering	

Security in Software Engineering	
	Secure Data Warehouses
<ul style="list-style-type: none"> <li>• Recommended Reading: <ul style="list-style-type: none"> <li>▪ [JCR 5] Fernández-Medina, E., Trujillo, J., Villarroel, R., and Mario Piattini (2006). Access Control and Audit Model for the Multidimensional Modeling of Data Warehouses. <i>Decision Support Systems</i>, 42, 1270-1289.</li> <li>▪ [JCR 7] Fernández-Medina, E., Trujillo, J., Villarroel, R., and Piattini, M. (2007). Developing Secure Data Warehouses with a UML extension. <i>Information Systems</i>, 32 (6), 826-856</li> <li>▪ [JCR 12] Fernández-Medina, E., Trujillo, and Piattini, M. (2007). Model Driven Multidimensional Modeling of Secure Data Warehouses. <i>European Journal of Information Systems</i>, 16, 374-389</li> <li>▪ [JCR 14, JCR 16, JCR 17, JCR18]</li> <li>▪ Papers from Priebe and Pernul about OLAP security.</li> </ul> </li> </ul>	
86	
Eduardo Fernández-Medina – Security in Software Engineering	

**Security in Software Engineering**  
Secure Data Warehouses

- Bill Inmon: A **data warehouse** is a subject-oriented, integrated, time-variant and non-volatile **collection of data** in support of management's **decision making process**
- Ralph Kimball: A **data warehouse** is a copy of transaction data specifically **structured for querying** and reporting

87

Eduardo Fernández-Medina – Security in Software Engineering


**Security in Software Engineering**  
Secure Data Warehouses

- DWs usually store **important** business information  
→ **sensitive** information
  - In some cases this information can be **protected** by local **laws** (personal data)
  - We are worried by “**confidentiality**”
- There are some **tools** for managing DW, which support security measures, but...
  - Once the DW **has been designed**
- If we incorporate **security** from the **beginning** of the DWs development, we **improve** the **quality**, and we **reduce** the **cost** (remember what we said some slices before)

88

Eduardo Fernández-Medina – Security in Software Engineering

**Security in Software Engineering**  
Secure Data Warehouses




- DWs need to be **designed**. Typical design stages are as follows:
  - **Multidimensional** design, where a conceptual model is developed.
  - **Logical** design, where a more concrete model is developed
  - **Physical** design, where the logical model is implemented in a particular platform (DBMS or OLAP tool)
- Our idea:
  - Chose an appropriate **notation** for modeling DWs
  - Define (at requirement level) a generic **security model** (MDS)
  - **Merge** that generic security model with the DW model
    - We then will model **secure DWs**
  - Obtain by automatic (or semi) **transformation** the implementation in our target platform
  - Have the possibility of applying **reverse engineering**

89

Eduardo Fernández-Medina – Security in Software Engineering

**Security in Software Engineering**  
Secure Data Warehouses



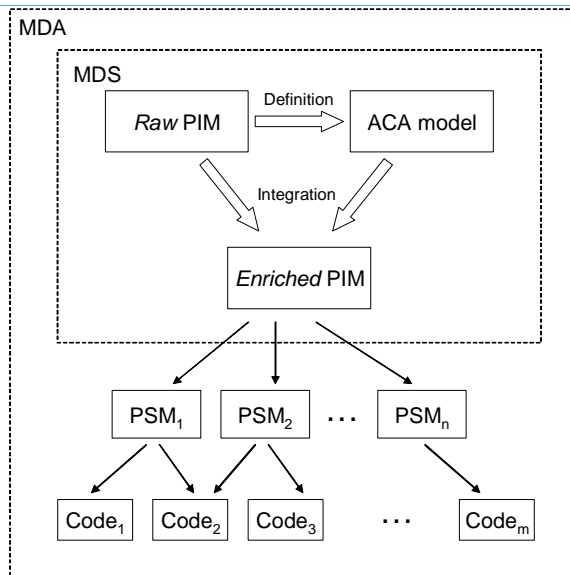
The diagram illustrates the design stages of a Secure Data Warehouse. It shows a flow from left to right across three time points: T1, T2, and T3. At T1 (Business Model), there is a box labeled CIM. An arrow points from CIM to a box labeled PIM at T2 (Conceptual Level). From PIM, two arrows branch out to two boxes labeled PSM<sub>1</sub> and PSM<sub>n</sub> at T3 (Logical Level). From each PSM box, two arrows branch out to two boxes labeled Code<sub>1</sub> and Code<sub>n</sub> at T3 (Code Level). Vertical dashed lines separate the stages: T1, T2, and T3.

90

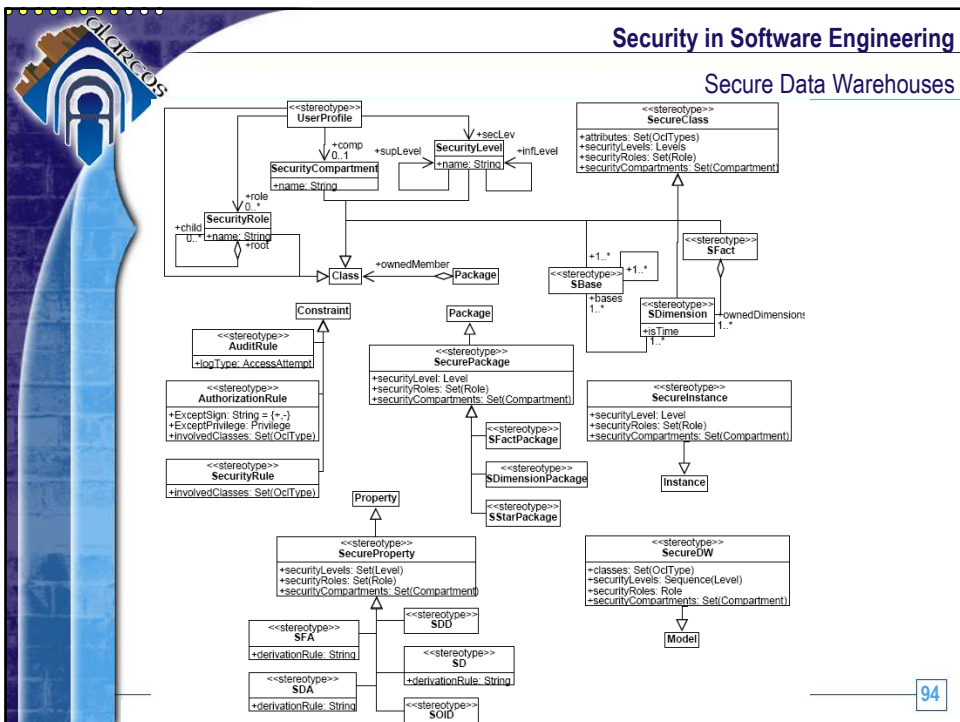
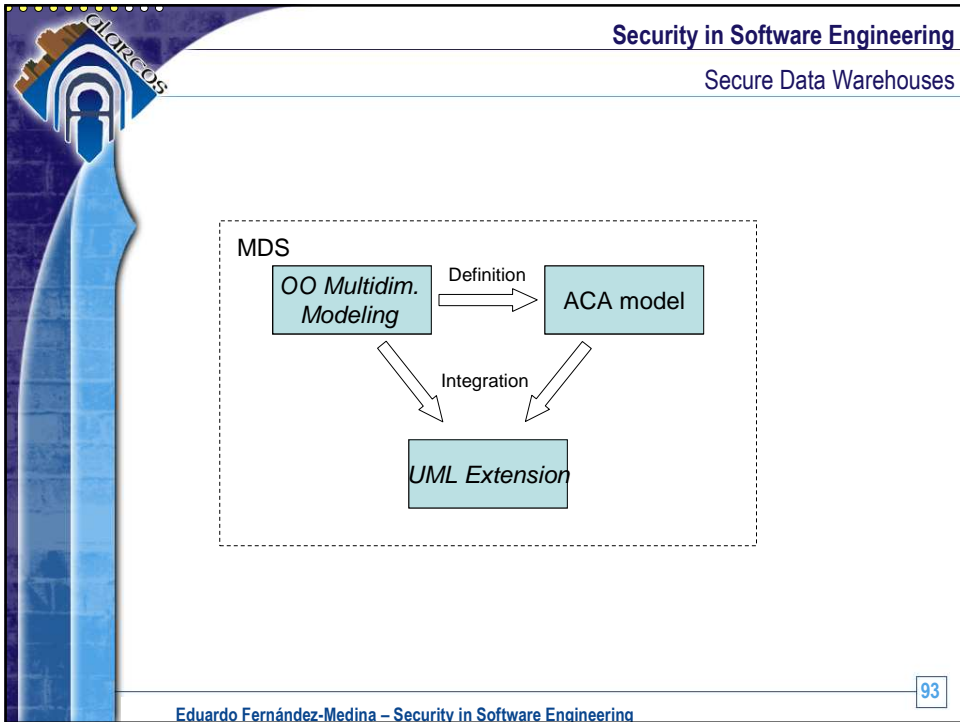
Eduardo Fernández-Medina – Security in Software Engineering

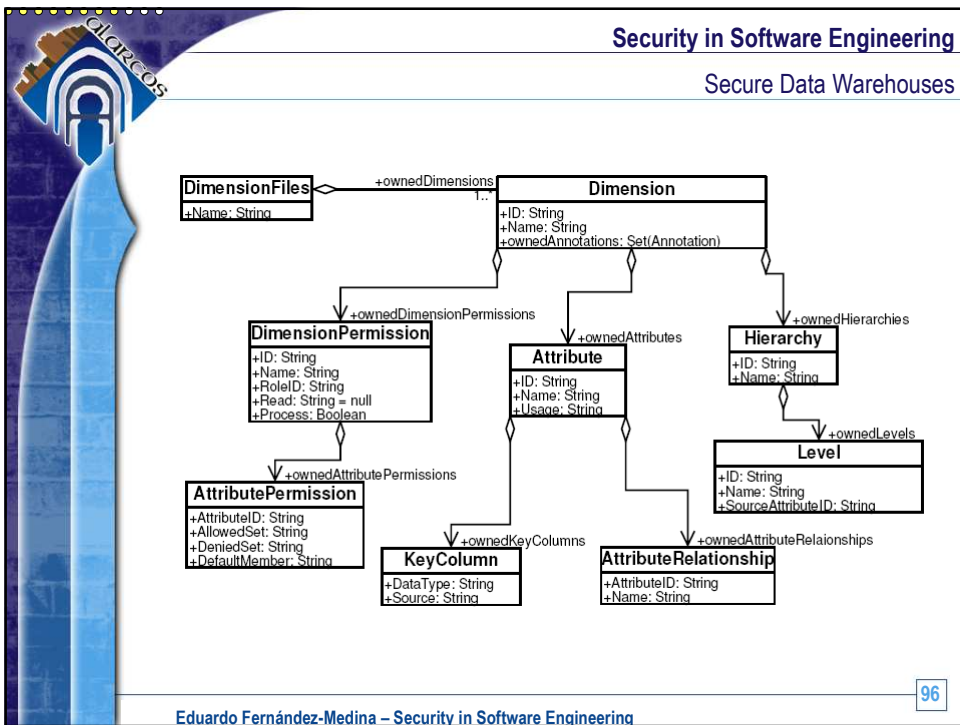
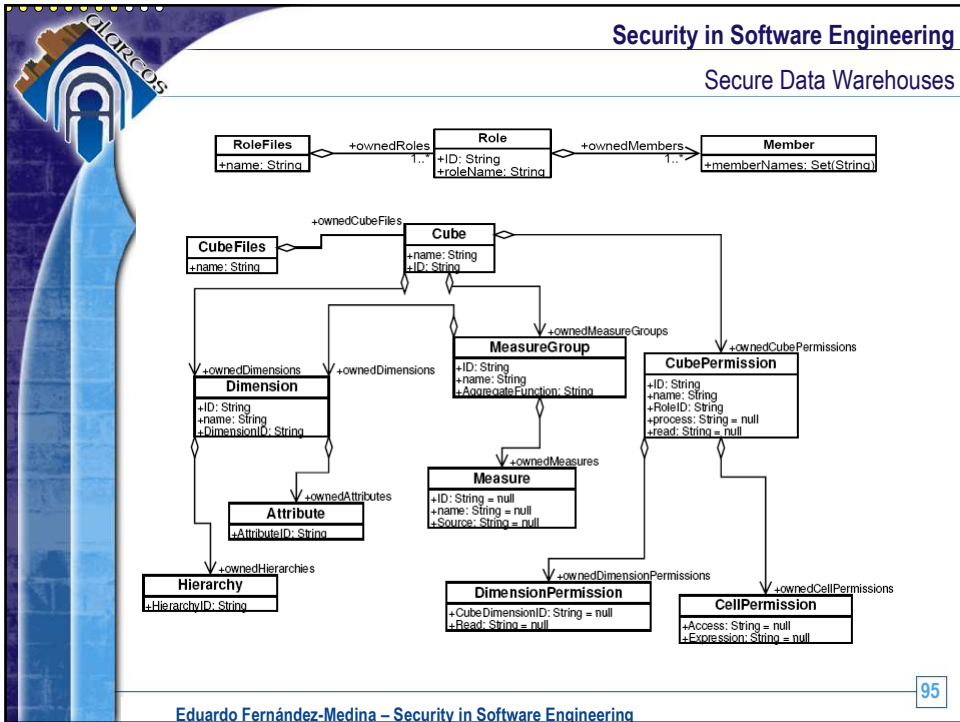
- Research activities:
  - Decide about the **DW paradigm** to incorporate security
    - Trujillo et al.
  - Study what **security aspects** can/need to be specified
    - Confidentiality
  - Define **metamodels** at different abstraction levels
    - Conceptual and Logical
  - Define **transformations** between these metamodels
  - Chose different **target platforms**, and study their security possibilities
    - Oracle, Pentaho, **SQL Server Analysis Services**
  - Define transformation for generating **code** for these **target platforms**

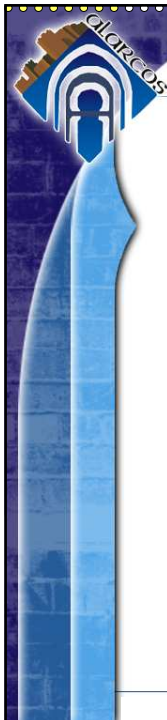
91



92



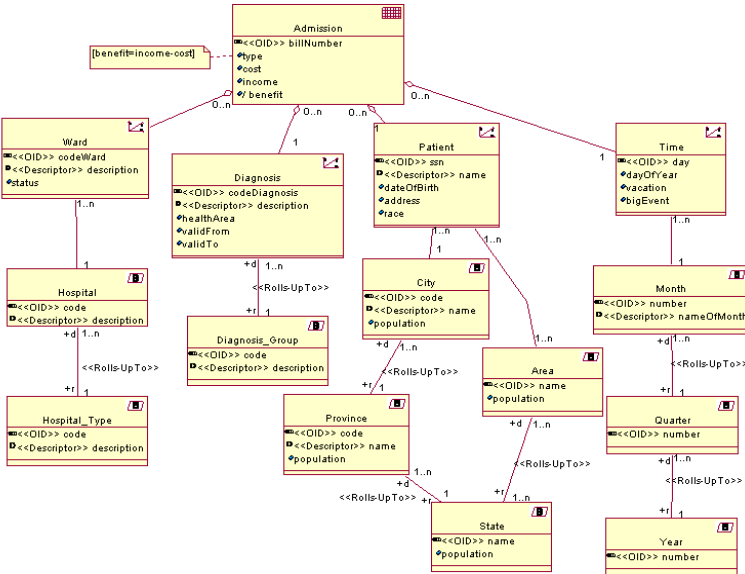
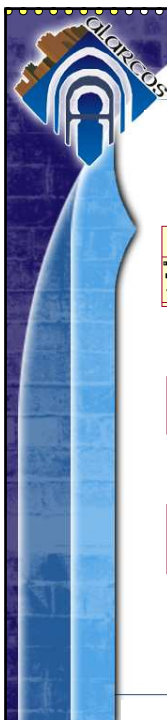
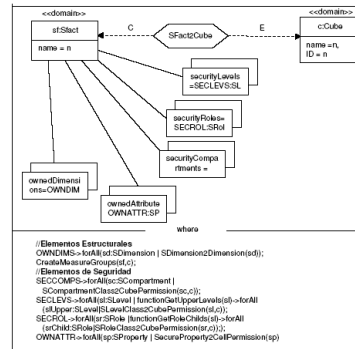
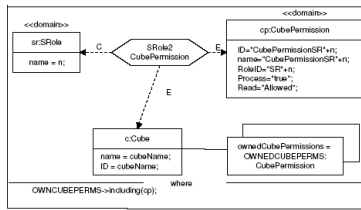


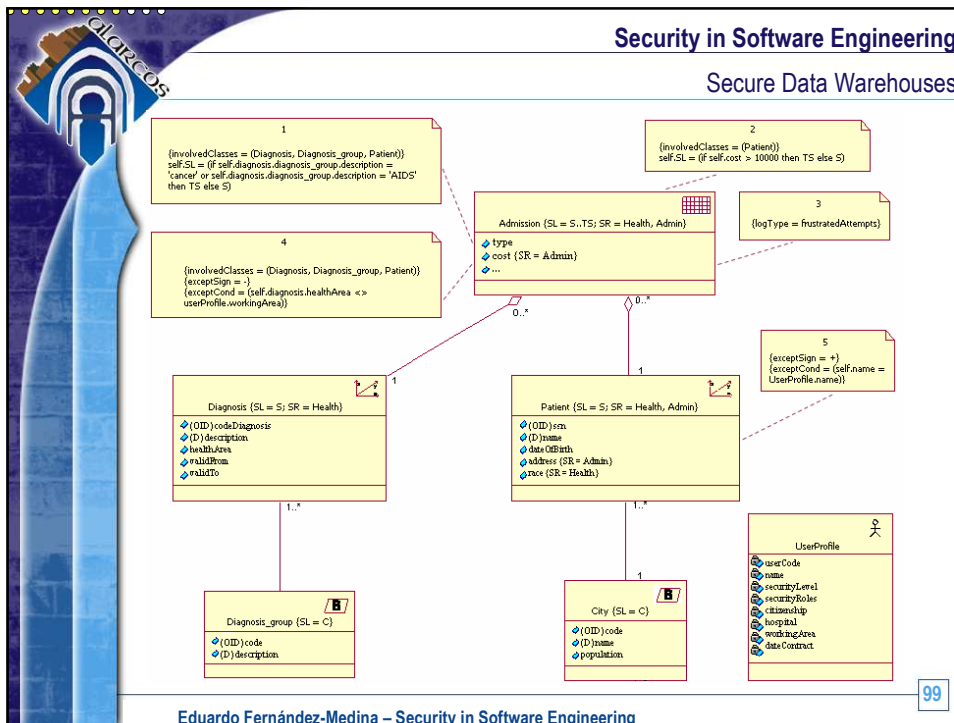


```

relation SFact2Cube{...}
relation CreateMeasureGroups{...}
relation Property2Measure{...}
relation SDimension2Dimension{...}
relation ProcessSBase{...}
relation CreateOwnedHierarchies{...}
relation SProperty2Attribute {...}

relation SCompartmentClass2CubePermission{...}
relation SRoleClass2CubePermission{...}
relation SLevelClass2CubePermission{...}
relation SCompartmentAtt2CellPermission{...}
relation SRoleAtt2CellPermission{...}
relation SLevelAtt2CellPermission{...}
}
  
```





**Security in Software Engineering**  
Content

- Introduction
- Secure IS Development – General Frameworks
- Security Requirements Engineering
- Security Architectural Patterns
- Model Driven Development
- Model Driven Security
- Secure Databases
- Secure Data Warehouses
- **Secure Business Process Models**
- Conclusions
- Events

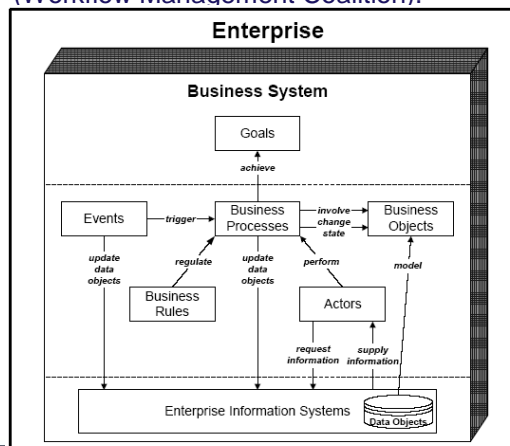
100

Eduardo Fernández-Medina – Security in Software Engineering

- Recommended Reading:
  - [JCR 13] A. Rodriguez, E. Fernandez-Medina, M. Piattini, An MDA Approach to Develop Secure Business Processes through a UML 2.0 Extension, *Computer Systems, Science and Engineering* 22 (5) (2007)
  - [JCR 10] A. Rodriguez, E. Fernandez-Medina, M. Piattini, A BPMN Extension for the Modeling of Security Requirement in Business Processes, *IEICE Transactions on Information and Systems* E90-10 (4), 745-752
  - [CIN 66] A. Rodriguez, E. Fernández-Medina, M. Piattini, Towards CIM to PIM Transformation: From Secure Business Processes Defined in BPMN to Use-Cases, in: Proceedings of the International Conference on Business Process Management, Brisbane, Australia, 2007, pp. 408-415.
  - [CIN 65] A. Rodriguez, E. Fernandez-Medina, M. Piattini, Analysis-Level Classes from Secure Business Processes Through Model Transformations, in: Proceedings of the International Conference on Trust, Privacy and Security in Digital Business, Regensburg, Germany, 2007, pp. 104-114.

101

- A **business process** is a set of **activities** or procedures which fulfill the high scale **goals**, in the context of and **organizational** structure, and defining roles, functions and relationships (Workflow Management Coalition).

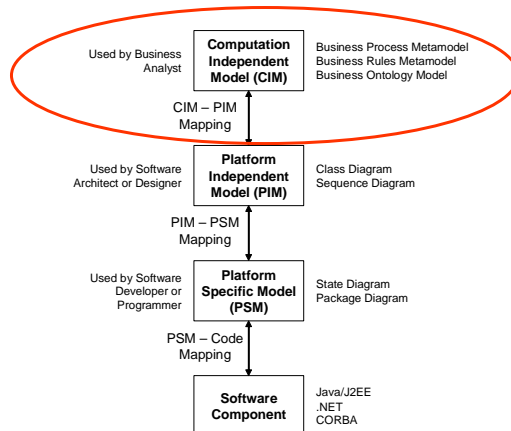


Barrios y Montilva

102

- Business processes describe business **functions** and the necessary **resources**.
- Business processes define the **methods** in which enterprises reach their **objectives**.
- Business processes models contain details about **what and how is done** within the company (**requirements**), and they frequently are the starting point for the **software development**.
  - They are a rich source of **requirements** of software that will implement these processes.
- Therefore, modeling business process is an **interesting** area of **software engineering**, due to it is placed in the **earliest** stages of the software development.

- Business processes can be integrated into a MDA architecture.



Security in Software Engineering  
Secure Business Process Models

- Therefore, our idea is to incorporate **security** requirements into **business processes**, and then, integrate business process into an **MDA architecture** for obtaining more concrete models that can be useful for **developing** information systems, and where security is an important aspect.
- There are not much experience expressing security requirements in so high **abstraction** levels, but empirical studies confirm that **business analyst** are able to express their **security** needs (at a very high abstraction level)

105

Eduardo Fernández-Medina – Security in Software Engineering

Security in Software Engineering  
Secure Business Process Models

- Abstract **Taxonomy** of security requirements

The diagram illustrates an abstract taxonomy of security requirements. It is structured as follows:

- Security** (Quality Factor)
  - Access Control
    - Identification
    - Authentication
  - Attack/Harm Detection
  - Non Repudiation
  - Integrity
    - Authorization
    - Data Integrity
    - Hardware Integrity
    - Personnel Integrity
    - Software Integrity
      - Immunity
  - Security Auditing
  - Physical Protection
  - Privacy
    - Anonymity
    - Confidentiality
  - Recovery
  - Prosecution

106

Eduardo Fernández-Medina – Security in Software Engineering

Security in Software Engineering  
Secure Business Process Models

## Notations for modeling Business Processes

Flow chart

Data flow diagrams

Entity-relationship diagrams

State-transition diagrams

Gantt Chart

Role activity diagrams

IDEF family techniques

Simulation techniques

Based on knowledge techniques

Petri Nets

Workflows techniques

UML 2.0  
Activity Diagram

Business  
Process

BPMN  
Business Process Diagram

107

Eduardo Fernández-Medina – Security in Software Engineering

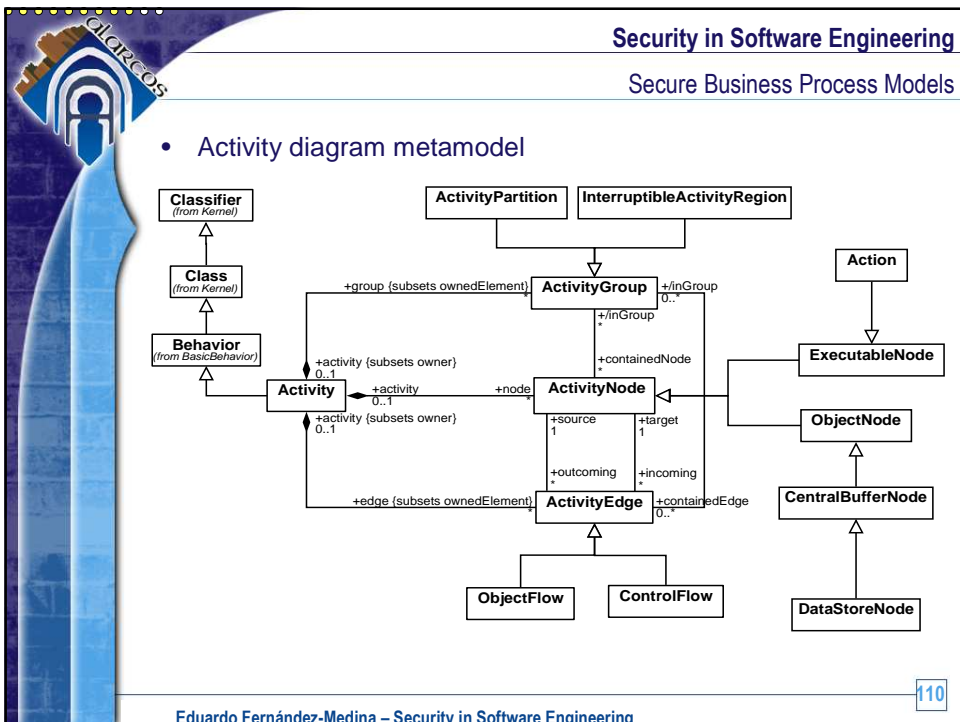
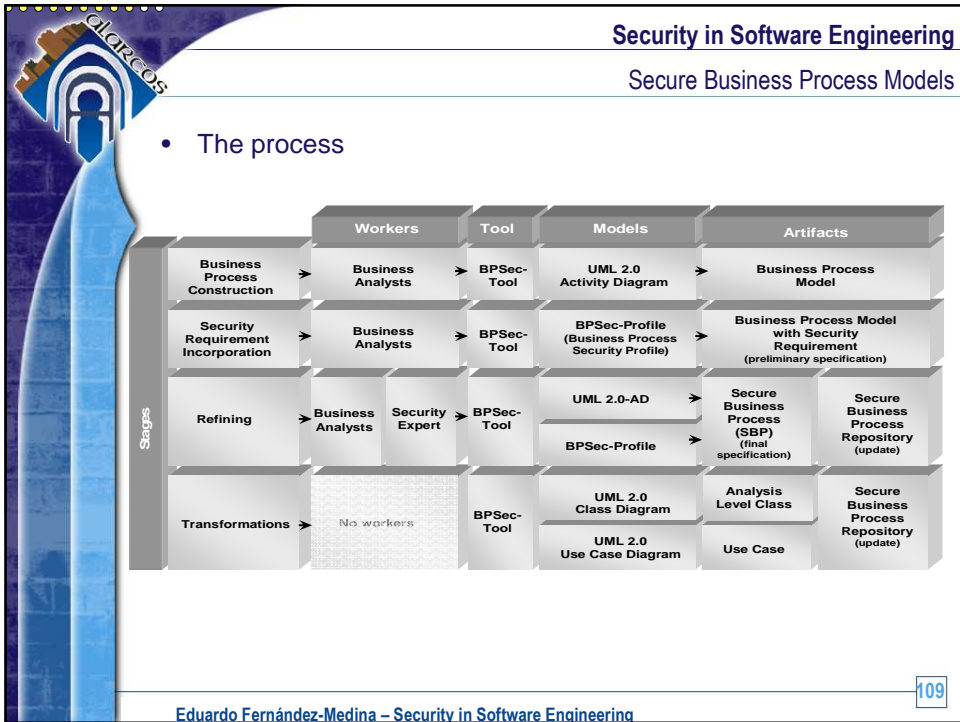
Security in Software Engineering  
Secure Business Process Models

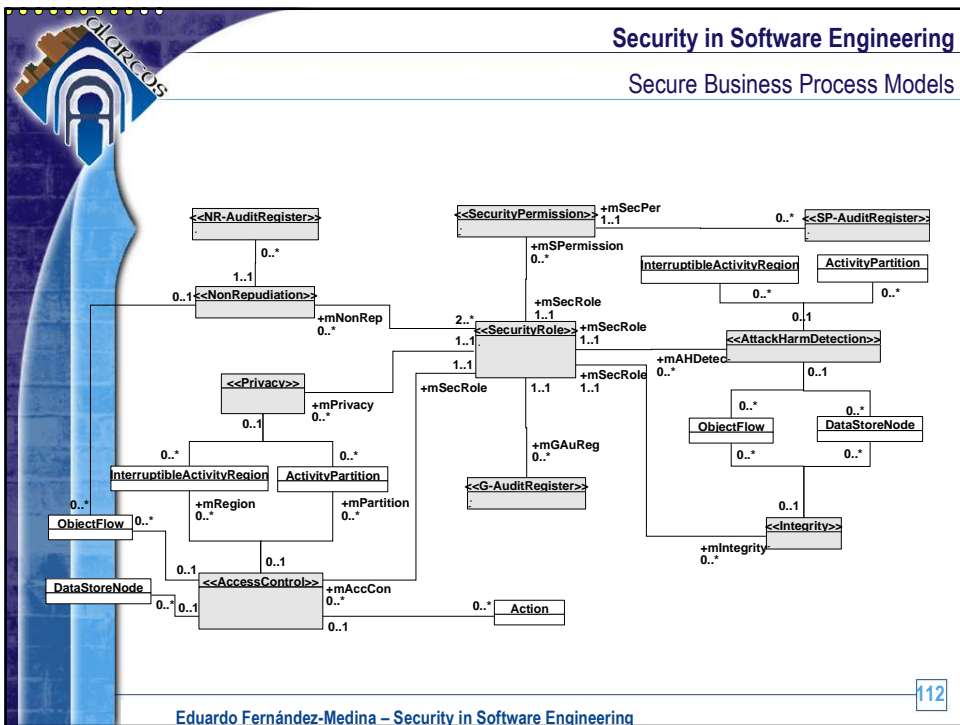
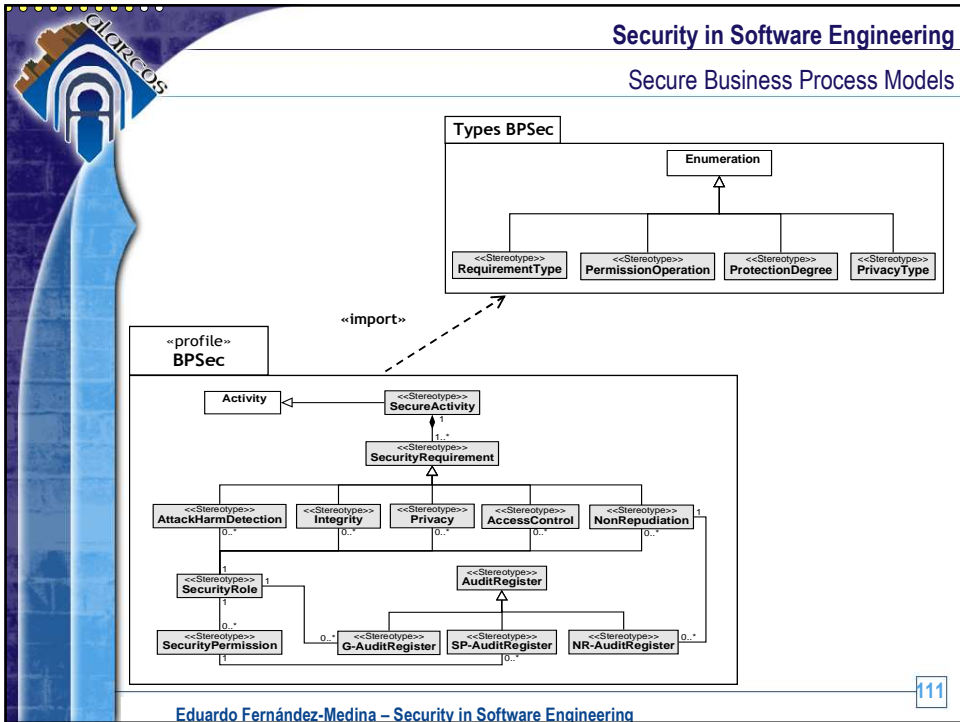
- Our Approach

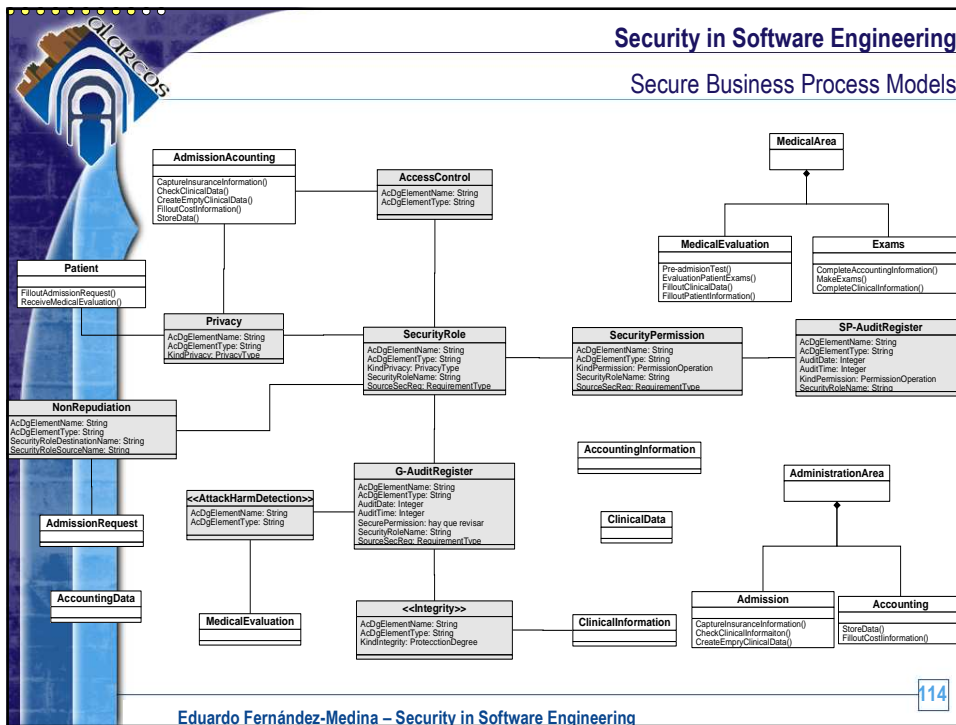
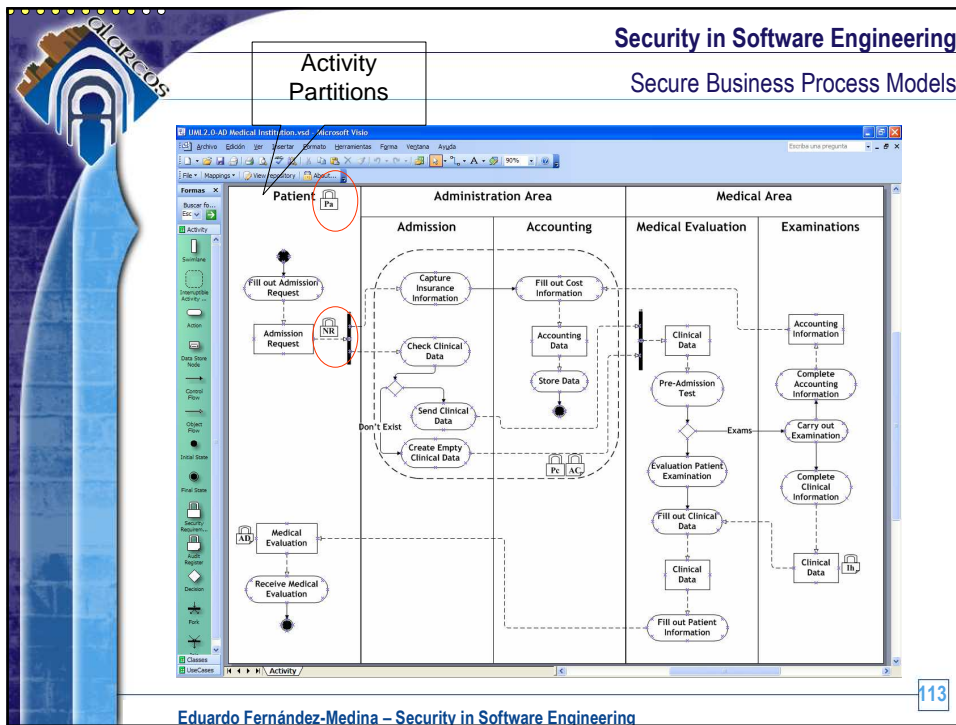
Model Driven Architecture	Our Proposal			Unified Process (disciplines)
Computation Independent Model	UML 2.0-AD	Secure Business Process Model	MBPSEC	Business Modeling
	BPsec-Profile			
Platform Independent Model	CIM2PIM transformations			Requirement Analysis & Design
	Analysis-level Class	Use Case		

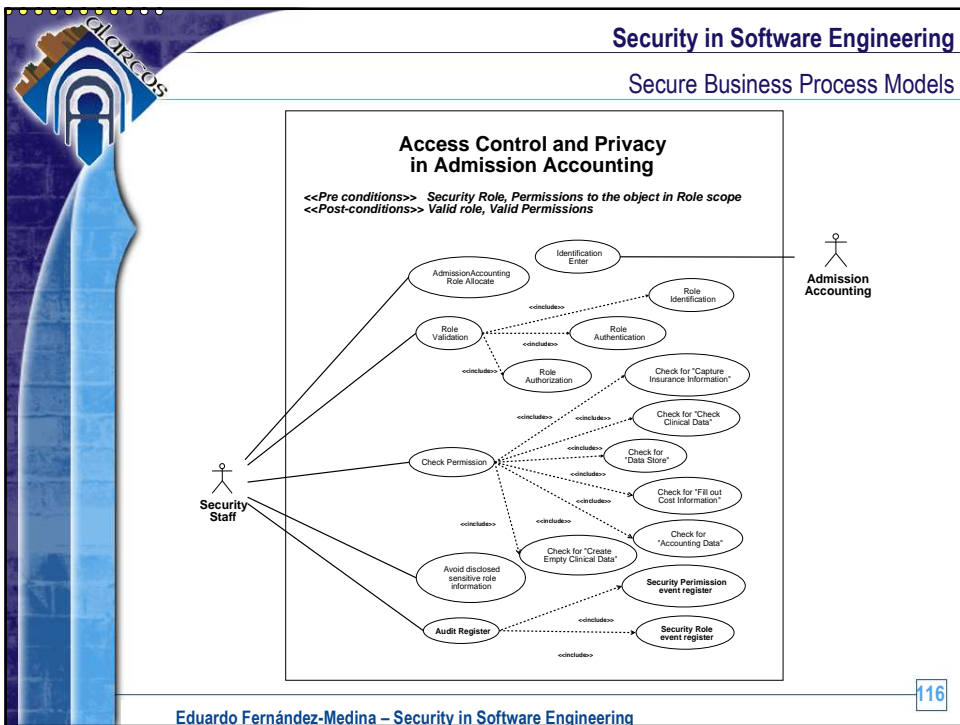
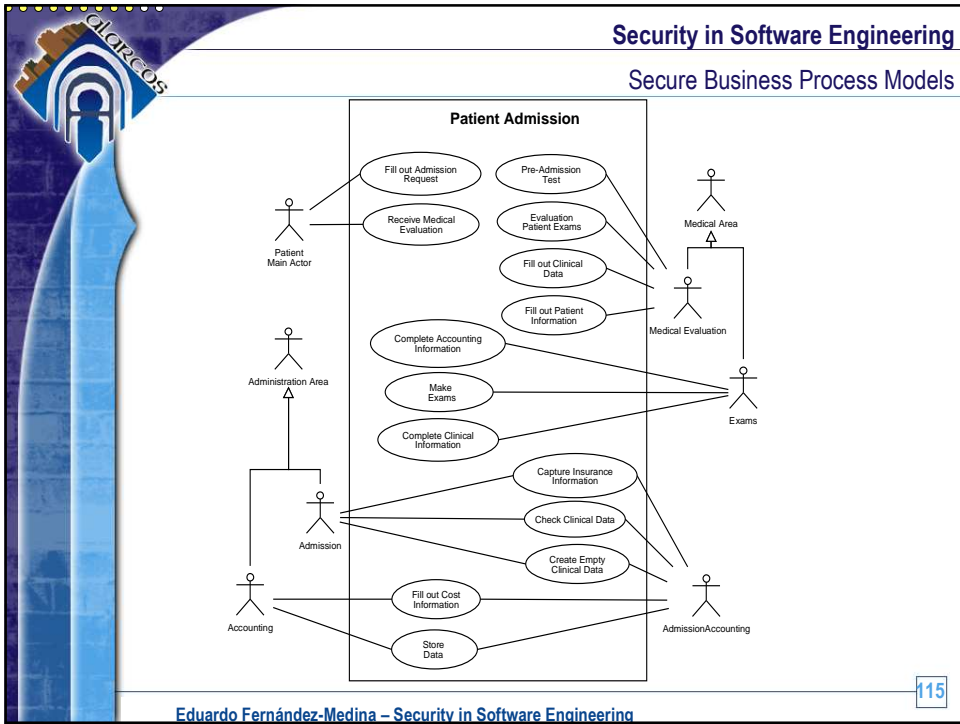
108

Eduardo Fernández-Medina – Security in Software Engineering












**Security in Software Engineering**  
Content

- Introduction
- Secure IS Development – General Frameworks
- Security Requirements Engineering
- Security Architectural Patterns
- Model Driven Development
- Model Driven Security
- Secure Databases
- Secure Data Warehouses
- Secure Business Process Models
- **Conclusions**
- Events

117

Eduardo Fernández-Medina – Security in Software Engineering




**Security in Software Engineering**  
Conclusions

- **Security** in **software engineering** is an **open** research area, with has evolved in the last years, but there are a huge number of **unresolved** topics.
- Security should definitively be integrated into the software development process as a **critical requirements**, due to the increasing **dependency** the current society has from information systems and technologies.
  - At this moment, is difficult to imagine a social or business environment where information systems and technologies are not critical.
- Model Driven Development is an interesting and **promising** software development **approach**, so maybe could be interesting incorporate security into **high level models**...

118

Eduardo Fernández-Medina – Security in Software Engineering



**Security in Software Engineering**

Conclusions

- If we correctly define **metamodels**, and we establish the correct **connections** between **models** and with the target **platforms**, we will be able to be close of the **automatic** software development, including **security** solutions for information systems.
- The scientific community **needs** researchers who intensify this research area, while other security technical research areas are overloaded.
  - Techniques for representing security requirements
  - New models, patterns, extensions, etc. for representing security details
  - Transformation specifications
  - Verification techniques
  - Methodologies for developing secure information systems and artifacts
  - CASE tools which support these techniques, methods, models...
  - Etc., etc., etc., etc....

119

Eduardo Fernández-Medina – Security in Software Engineering



**Security in Software Engineering**

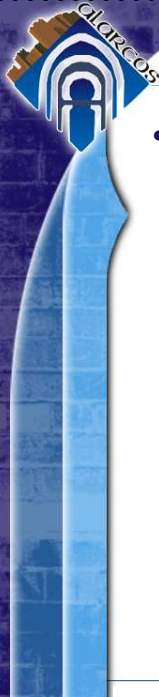
Conclusions

- Some **topics**:
 

<ul style="list-style-type: none"> <li>▪ Methodologies which incorporate security modeling</li> <li>▪ Business Process Security</li> <li>▪ Security Requirements Engineering</li> <li>▪ Security requirements</li> <li>▪ Modeling security requirements</li> <li>▪ Languages for modeling security</li> <li>▪ Model-driven security engineering</li> <li>▪ Secure architecture and design</li> <li>▪ Patterns for security modeling</li> <li>▪ Secure implementation</li> <li>▪ Testing for security</li> <li>▪ Model-based testing for security</li> <li>▪ Verification and validation techniques for security properties</li> <li>▪ Tool support for modeling security</li> <li>▪ Standards for Security</li> </ul>	<ul style="list-style-type: none"> <li>▪ Security for Grid computing</li> <li>▪ Security for Mobile Computing</li> <li>▪ Web Services Security</li> <li>▪ Security for Databases and Data Warehouses</li> <li>▪ Metrics for Security</li> <li>▪ Security in agile software development</li> <li>▪ Static and dynamic analysis for security</li> <li>▪ Aspect-oriented software development for secure software</li> <li>▪ Security and usability</li> <li>▪ Reverse engineering security models</li> <li>▪ Security-oriented software reconfiguration and evolution</li> <li>▪ Automated development</li> </ul>
---	---

120

Eduardo Fernández-Medina – Security in Software Engineering




**Security in Software Engineering**

Conclusions

- Finally....
  - Scientific community wants a “Security Ontology” and a “**Security Manifesto**” which declare the need of defining a Secure Software Engineering discipline which make the development of secure software as an engineering, rather than an art.

121

Eduardo Fernández-Medina – Security in Software Engineering



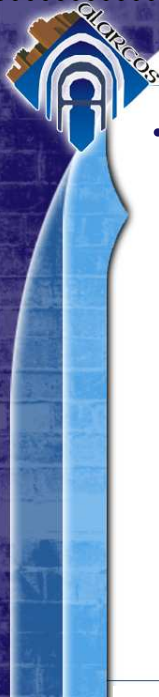
**Security in Software Engineering**

Content

- Introduction
- Secure IS Development – General Frameworks
- Security Requirements Engineering
- Security Architectural Patterns
- Model Driven Development
- Model Driven Security
- Secure Databases
- Secure Data Warehouses
- Secure Business Process Models
- Conclusions
- **Events**

122

Eduardo Fernández-Medina – Security in Software Engineering



**Security in Software Engineering**  
Events

- Almost all software engineering conferences include “security” into their important topics, or even as specific tracks or specialized workshops.
  - ICSE: International Conference on Software Engineering
  - ICSEA: International Conference on Software Engineering Advances
  - SIGSOFT: Foundations of Software Engineering
  - ASE: Automated Software Engineering
  - Etc.

123

Eduardo Fernández-Medina – Security in Software Engineering



**Security in Software Engineering**  
Events

- Some specific events related to security in software engineering:
  - **International Workshop on Security in Information Systems (WOSIS, in association with ICEIS)**
    - Papers: February
    - Funchal, Madeira, Portugal. 8-12 June 2010
    - <http://www.iceis.org/> (cfp to be announced)
      - Prestigious committee program
      - Prestigious keynote speaker
      - Journal link for publishing best papers

**Please, submit here your papers**

124

Eduardo Fernández-Medina – Security in Software Engineering



**Security in Software Engineering**  
Events

- Some specific events related to security in software engineering:
  - Modeling Security Workshop 2008 (In Association with MODELS)
    - Toulouse, France. 28 September 2008
    - <http://www.modelsconference.org/>
  - International Symposium on Engineering Secure Software and Systems – ESSoS 2010
    - Abstract: 15 September
    - Paper: 30 September
    - Pisa, Italy, 3-4 February 2010
    - <http://distrinet.cs.kuleuven.be/events/essos2010/>
  - International Conference on Availability, Reliability and Security (ARES 2010)
    - Paper: 1 September
    - Krakow, Poland. 15-18 February 2010
    - <http://www.ares-conference.eu/conf/index.php/call-for-papers>

125

Eduardo Fernández-Medina – Security in Software Engineering



**Security in Software Engineering**  
Events

- Some specific events related to security in software engineering:
  - Third International Workshop on Secure Software Engineering 2009 (In association with ARES)
    - Paper: 30 September
    - Krakow, Poland. 15-18 February 2010
    - <http://sintef.org/Home/Information-and-Communication-Technology-ICT/Software-Engineering-Safety-and-Security/Projects/SecSE-2010/>
  - International Workshop on Software Engineering for Secure Systems 2010 (In association with ICSE)
    - Paper: January
    - <http://homes.dico.unimi.it/~monga/sess09.html>

126

Eduardo Fernández-Medina – Security in Software Engineering



**Thank you for your  
attention**



**Alarcos Research Group**  
<http://alarcos.inf-cr.uclm.es>  
[Eduardo.FdezMedina@uclm.es](mailto:Eduardo.FdezMedina@uclm.es)  
<http://alarcos.inf-cr.uclm.es/per/efmedina/>